NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

RAPIDLY DEPLOYABLE MOBILE SECURITY SOLUTION

by

Liam J. Dorney
Travis C. Miller

March 2016

Thesis Advisor:	Man-Tak Shing
Second Reader:	Arijit Das

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE	*Form Approved OMB No. 0704–0188*

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington, DC 20503.

1. AGENCY USE ONLY ***(Leave blank)***	**2. REPORT DATE** March 2016	**3. REPORT TYPE AND DATES COVERED** Master's thesis

4. TITLE AND SUBTITLE RAPIDLY DEPLOYABLE MOBILE SECURITY SOLUTION	**5. FUNDING NUMBERS** **ACCT: 9761769** **JON: RCJ20**
6. AUTHOR(S) Liam J. Dorney and Travis C. Miller	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943-5000	**8. PERFORMING ORGANIZATION REPORT NUMBER**
9. SPONSORING /MONITORING AGENCY NAME(S) AND ADDRESS(ES) Space And Naval Warfare Systems Command 53560 Hull ST, San Diego, CA 92512	**10. SPONSORING / MONITORING AGENCY REPORT NUMBER**

11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government. IRB protocol number ____N/A____.

12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release; distribution is unlimited	**12b. DISTRIBUTION CODE** A

13. ABSTRACT (maximum 200 words)

The Navy has seen a significant increase in the presence of mobile and smart devices on its units due to advancements in technology and younger sailors' desire to be connected at all times. These devices create security threats due to their easily concealable size and their host of connectivity and image related features. The insider threat (intentional or not) now includes the ability to take photos, record conversations, share data wirelessly, and communicate official use and classified information, all more easily than ever before.

Current enterprise solutions and associated policy does not address managing personal devices. In fact, management of personal devices is currently outside the Department of Defense (DOD) effort to control Personal Electronic Devices (PED) since the organization does not own the device and therefore has no way to mandate what must or must not be installed on them. The current path to a bring your own device (BYOD) policy is unclear. Security vulnerabilities with these devices have not been addressed in a uniform matter in policy or in practice. It is with these statements in mind that we address how to take the first steps in developing feasible management of personal devices on naval units and potentially throughout the DOD.

In this thesis, we provide a thorough evaluation of National Institute for Standards and Technology, Defense Information Systems Agency, and DOD publications to provide a starting point for adapting current policy and to guide the development of our application. We then examine the feasibility of implementable software application solutions to hardware features that pose a threat to security. Specific research addresses why each hardware feature on a mobile device is a security concern, how it is controlled inside the Android Studio API, and how we utilize these controls to lockdown and then unlock said hardware features through a simple proof of concept Android application. Finally, we provide examples of how future work can grow our application into a security-manager controlled program to secure devices and find a path toward making BYOD a reality.

14. SUBJECT TERMS mobile device, mobile security, android application program, bring your own device (BYOD), cyber security, cyber policy, insider threat, U.S. Navy cyber policy, risk management	**15. NUMBER OF PAGES** 205
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT Unclassified	**18. SECURITY CLASSIFICATION OF THIS PAGE** Unclassified	**19. SECURITY CLASSIFICATION OF ABSTRACT** Unclassified	**20. LIMITATION OF ABSTRACT** UU

THIS PAGE INTENTIONALLY LEFT BLANK

RAPIDLY DEPLOYABLE MOBILE SECURITY SOLUTION

Liam J. Dorney
Lieutenant, United States Navy
B.S., University of Idaho, 2009
M.S., University of Idaho, 2009

Travis C. Miller
Lieutenant, United States Navy
B.A., University of Tennessee, 2003

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN COMPUTER SCIENCE

from the

NAVAL POSTGRADUATE SCHOOL
March 2016

Approved by: Man-Tak Shing
Thesis Advisor

Arijit Das
Second Reader

Peter Denning
Chair, Department of Computer Science

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

The Navy has seen a significant increase in the presence of mobile and smart devices on its units due to advancements in technology and younger sailors' desire to be connected at all times. These devices create security threats due to their easily concealable size and their host of connectivity and image related features. The insider threat (intentional or not) now includes the ability to take photos, record conversations, share data wirelessly, and communicate official use and classified information, all more easily than ever before.

Current enterprise solutions and associated policy does not address managing personal devices. In fact, management of personal devices is currently outside the Department of Defense (DOD) effort to control Personal Electronic Devices (PED) since the organization does not own the device and therefore has no way to mandate what must or must not be installed on them. The current path to a bring your own device (BYOD) policy is unclear. Security vulnerabilities with these devices have not been addressed in a uniform matter in policy or in practice. It is with these statements in mind that we address how to take the first steps in developing feasible management of personal devices on naval units and potentially throughout the DOD.

In this thesis, we provide a thorough evaluation of National Institute for Standards and Technology, Defense Information Systems Agency, and DOD publications to provide a starting point for adapting current policy and to guide the development of our application. We then examine the feasibility of implementable software application solutions to hardware features that pose a threat to security. Specific research addresses why each hardware feature on a mobile device is a security concern, how it is controlled inside the Android Studio API, and how we utilize these controls to lockdown and then unlock said hardware features through a simple proof of concept Android application. Finally, we provide examples of how future work can grow our application into a security-manager controlled program to secure devices and find a path toward making BYOD a reality.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF ACRONYMS AND ABBREVIATIONS

AS	Android Studio
BNRG	Big Nerd Ranch Guide
BYOD	bring your own device
CANES	consolidated afloat networks and enterprise services
CAT	category
CIO	chief information officer
CMD	commercial mobile device
CMI	classified message incident
CNSS	Committee on National Security Systems
CO	commanding officer
CTTA	Certified TEMPEST Technical Authority
DAA	designated approving authority
DARPA	Defense Advanced Research Projects Agency
DHS	Department of Homeland Security
DISA	Defense Information Systems Agency
DMD	Developers Material Design
DOD	Department of Defense
DOJ	Department of Justice
DON	Department of the Navy
DONCIO	Department of the Navy Chief Information Officer
FBI	Federal Bureau of Investigation
FIPS	Federal Information Processing Standards
FISMA	Federal Information Security Management Act
GAD	Google Android Development
GFE	government furnished equipment
IDE	interactive development environment
IMEI	international mobile equipment identifier
IMSI	international mobile subscriber identifier
IO	input/output
IT	information technology

MAS	mobile application store
NFC	near field communication
NIST	National Institute of Standards and Technology
ODNI	Officer of the Director of National Intelligence
OS	operating system
PED	personal electronic device
QR	quick response
RFID	radio frequency identification
RMF	Risk Management Framework
SIPRNET	secret intent protocol routing network
SWLAN	secure WLAN
SP	special publication
SPAWAR	Space and Naval Warfare Systems Command
SRG	security requirement guides
STIG	Security Technical Implementation Guide
TPM	technical performance measurement
VM	virtual machine
WLAN	wireless local area network

ACKNOWLEDGMENTS

We cannot say enough about the support we have received while working on this thesis. Our families have put up with us being gone late in the evenings, our teachers have allowed us to incorporate our ideas into our class projects, and our advisors have guided us down this path.

To Man-Tak Shing and Arijit Das: thank you for taking the time to read over our thoughts and for helping us to make sense of them. Your guidance was instrumental in our completion of this project.

To Jerel Yam: we cannot say enough about how much you helped us in this project. You helped two programmers with very limited experience put their ideas to task. The time you spent helping us make sure all coding was correct was invaluable.

In addition, I want to thank my wife, Jenna Miller. She is a fantastic bride who has amazing patience and she enables me in all that I do. The time commitment she gave to supporting me was huge and appreciated. I could not ask for more from such an amazing wife and I would not have been able to do this without her.

—Travis Miller

I also would like to thank my wife, Jennifer; daughters, Hannah, Lily, and Evelyn; and son, Kol for their loving support. Their understanding through many long hours, late nights, and missed weekends made this success possible.

—Liam Dorney

THIS PAGE INTENTIONALLY LEFT BLANK

I. INTRODUCTION

A. PROBLEM

Naval platforms continue to have instances of individuals carrying unauthorized devices into restricted or secure spaces or using devices in ways that are not in line with naval regulations. Seeing a mobile device in spaces such as CIC or radio puts leadership in the difficult position of self-reporting the incident and handling the device. In addition to embarrassment to the command for having these instances occur, the presence of these devices presents an opportunity to maliciously photograph, record, or document data in a way that can be very difficult to detect. Even in instances without malintent this is a known violation of naval and Department of Defense (DOD) policy.

The use of these devices is not going away. Sailors today report to their command with more devices than ever in the form of computers, tablets, and smart phones. As technology progresses it is important that the fear of a security incident does not restrict successful implementation of mobile devices where appropriate and possible. So far, the Navy has avoided setting a clear bring your own device (BYOD) policy. Instead, devices are prohibited in their entirety or mismanaged to an extent that the written guidance has little meaning. Written restriction continues to be ineffective in limiting individuals carrying mobile devices throughout the ship. For this reason, it is necessary to establish what would be required to lock down the vulnerable features of non-enterprise procured mobile devices. An examination of characteristics that increase the possibility for security incidents along with proposed mitigators is discussed. We create an Android application to demonstrate what settings on a phone can be manipulated to lock down high-risk features (including the camera, microphone, Wi-Fi, Bluetooth, etc.).

The application's development is driven by information assurance requirements and known security concerns. Documentation exists within the Defense Information Systems Agency (DISA), across DOD organizations, and in the corporate world that outline security concerns associated with a mobile device's presence. These is used to develop the application, ensuring policy recommendations associated with mobile devices are implemented as possible. Additionally, examination of the current application marketplace and available tools as models are shown and referenced for a deeper application development.

The developed application's implementation must be simple and activation must be quick for the end users. With this in mind, the use of a Quick Response (QR) code and near field communication (NFC) technology is examined to activate the application, change security manager settings quickly, and implement a commander's device policy on board a naval ship. The use of QR codes or NFC also presents the opportunity to update the settings on a phone as it moves throughout a ship. For example, the back half of a submarine would have completely different device permissions (essentially fully locked down) when compared to the front half where blue tooth and voice-recorded notes on a device might be acceptable. Flexibility of implementation is a principle concern.

Further, in-depth policy evaluation is conducted alongside possibility of quickly adjusting the settings for so many end-user devices. Focusing on Android also allows access to data that will provide expected market reach for each operating system (OS) iteration based on the application's development. After providing the code for implementation, pre-programmed NFC devices are used to study how easily settings are changed and how quickly a security manager can push updates for new policy settings in dynamic environments such as those on a warship. All of this provides a basis to begin using the devices in sailors' possession in a manner that is acceptable for them and useful for the Navy.

B. RESEARCH QUESTIONS

Discussion of any BYOD policy will start with a commander asking how safe the device actually is. Defining what is meant by locking down a device, and how much can be controlled is the first step in setting the appropriate policy. Additionally, that definition must be compared with best security practices and current Navy policy to ensure that there is an acceptable level of comfort to security managers and commanding officers when implementing a mobile device policy. Clearly stating how device lockdown is defined according to the Navy, so that these devices represent less of a threat, and how it can be implemented afloat and ashore is the first goal.

The subsequent questions involve the device interaction themselves. Is the development of a basic mobile application with QR code or NFC interaction even possible for accessing those functions defined as dangerous in the lockdown definition? If so, what is an efficient example that can be implemented and expanded for Navy-wide use and if not, what recommendations can still be made for some form of BYOD implementation? With an application that is meeting the requirements, how are the various lockout and control features implemented and what form of data logging can be put into place in the future to ensure that users are not disabling the application on the end device? These questions will help establish the boundaries of an application and detail the areas that require further study and development.

Finally, once the features that can be locked down have been identified and an application demonstrates implementation, will this meet the Navy's and the DOD's requirements for a secure device? If not, what is keeping it from being a fully secure device and what expectations of security have been increased on the end user device? Additional future work is discussed to fill any gaps and help this app grow in its utility. In the same section, we try to answer what future study areas might help in the utilization of sailors' devices for Navy requirements. Additionally, we explore whether extra OS profiles be created or administrator apps be designed enabling sailors to access unsecure but official use only websites and databases or is the use of a personal device still too far off beyond simply limiting its features.

The utilization of smart, mobile devices has already been implemented at an enterprise level. The defining of requirements and approval of devices is not a fast process. The DOD is getting better at provisioning smart devices and providing them with a means of security already installed and demonstrating implementation options.

C. DEFINING THE STAKEHOLDERS

The stakeholders for a BYOD enabling application can be broken into three categories with overlaps between them. Those are the end users, leadership at individual commands, and the DOD/Navy as a whole. All of these have an interest in understanding smart usage of personal devices and avoidance of security incidents afloat. By investigating what the actual concerns associated with personal devices are and building software mitigations to lock down features, it may be possible to increase the sailors utilization of devices they already have while commanders get greater control over what can be running while on board.

1. The Sailors and Their Devices

Sailors are reporting to their command with more electronic devices than ever. The thought of having to exist in any environment without access to a mobile device is troublesome, as dependency on these devices had grown. Users now use smart devices to provide access to navigation while driving, data/educational content, entertainment, social media, and breaking news in addition to the traditional roles of phone calls and text messages (Figure 1).[1] This dependence leads sailors to believe their device must always be on them so that they are always connected. Recent studies indicate that individuals that regularly rely on a smart device feel increased levels of anxiety when they do not have access to it.[2] This creates a unique challenge for a Navy that is trying to keep official information from spilling out into the civilian world, reduce the number of security incidents on board a vessel, and still allow technology to have a place among our service members.

It is the combination of avoiding security incidents and sailors wanting access to their devices that is the rub. Should mobile devices be taken away once sailors step across the brow of a ship or should they be allowed to roam any space where they are assigned with a mobile hard drive, camera, and microphone attached to their hip? Has a consideration been made about the possibility for inadvertent recording of the spaces by

[1] Aaron Smith, *U.S. Smartphone Use in 2015* (Washington, DC: Pew Research Center 2015),http://www.pewinternet.org/2015/04/01/us-smartphone-use-in-2015/.

[2] Nathan Hurst, “IPhone Separation Linked to Physiological Anxiety, Poor Cognitive Performance, MU Study Finds,” *MU News Bureau*, January 8, 2015, http://munews.missouri.edu/news-releases/2015/0108-iphone-separation-linked-to-physiological-anxiety-poor-cognitive-performance-mu-study-finds/.

taking photos, the microphone being turned on and recording a conversation, or any of that data being pushed automatically to the cloud once outside the skin of the ship?

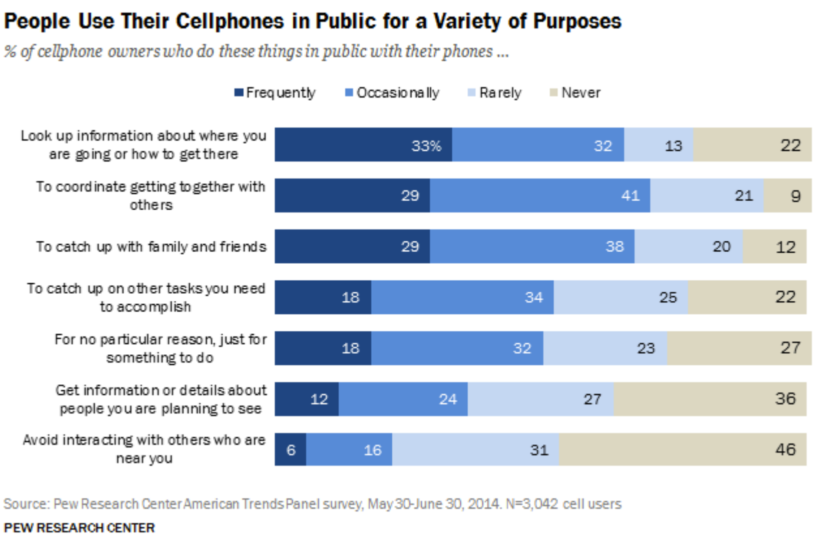
It is obvious that taking away or restricting cell phones would work to a degree, but as with secure facilities people would still accidentally carry them in. Additionally, service members' need to feel connected and within reach of family members, which leads to an increased desire to have the phone on hand. PEW research indicates not just a growing reliance on smart phones and devices for quick access to data, but shows that 46 percent of adults feel their cell phone is "something they can't imagine living without" (Figure 2).[3] A report from the same organization a year earlier has this sentiment attributed to 29 percent of the adult population.[4]

As mobile device capability grows, individual dependence on these devices will also grow. The myriad of activities and ways to connect provide a sense of comfort and access to the online world. Sailors will report to commands with these devices, which represent an unleveraged technology for the DOD. This has been recognized by attempts to fit a BYOD future into the organization. The topic then shifts to one in which we have a personally procured device that is secure enough to be carried inside our warships, locks down those components that allow recording, and leaves the device useful enough to provide some service. Access to data exchange can and should enable our sailors to complete routine, unclassified tasks and training without having to wait on access to limited computers on board a warship. Proving which mobile device features can be locked down and demonstrating the methods for doing this is the first step in this process.

[3] Smith, *U.S. Smartphone Use in 2015*.

[4] Pew Research Center, "Mobile Technology Fact Sheet," Pew Research Center, 2013, http://www.pewinternet.org/fact-sheets/mobile-technology-fact-sheet/.

Figure 1. American's rely on the smart devices for more than phone calls


People Use Their Cellphones in Public for a Variety of Purposes

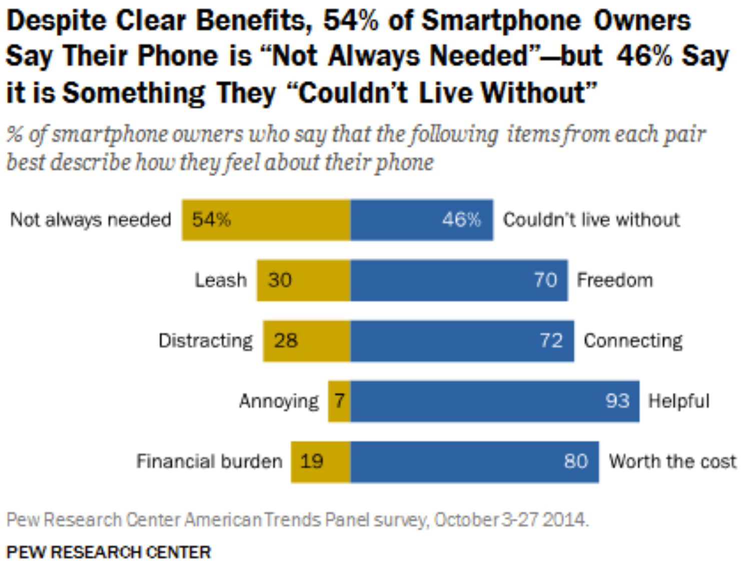
% of cellphone owners who do these things in public with their phones ...

Source: Pew Research Center American Trends Panel survey, May 30-June 30, 2014. N=3,042 cell users

PEW RESEARCH CENTER

The days of using a cell phone for a short message or phone call have long passed. American's now use smart phones as a line to the outside world for coordination of events, social media, and even as a GPS. Source: Pew Research Center, "People Use Their Cellphones in Public for a Variety of Purposes," Pew Research Center, August 25, 2015, http://www.pewinternet.org/2015/08/26/americans-views-on-mobile-etiquette/2015-08-26_alone-together_0_03/.

Figure 2. Every year more smartphone owners view their device as a necessity



Public opinion on cellphones as a social distraction has shifted from viewing them as something nice to have to a device that is necessary and connects them to the outside world. They are viewed as a helpful tool for daily life. Source: Aaron Smith, *U.S. Smartphone Use in 2015* (Washington, DC: Pew Research Center 2015), http://www.pewinternet.org/2015/04/01/us-smartphone-use-in-2015/.

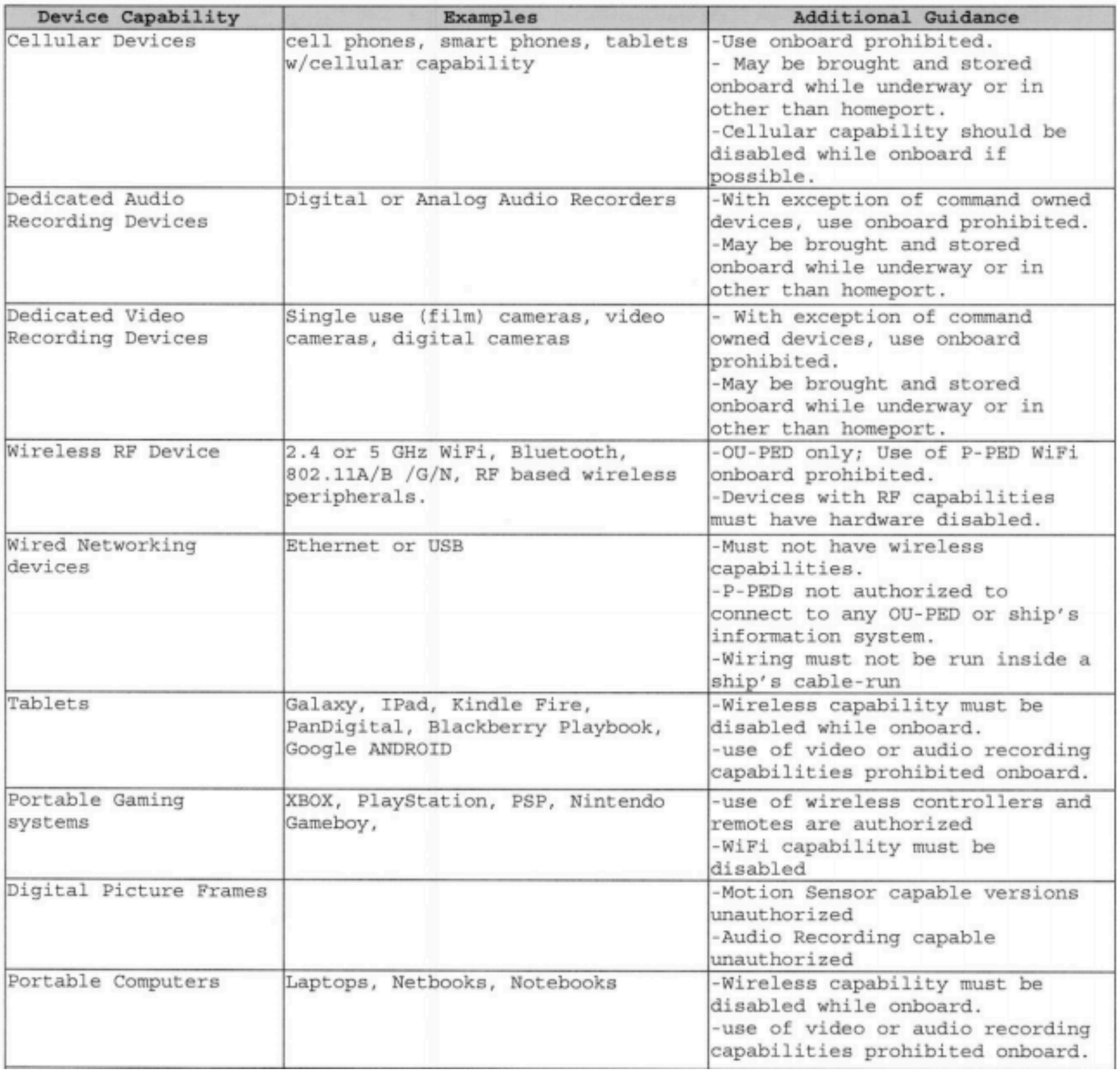
2. The Commander's Policy and Implementations

Leadership on the waterfront is left with a myriad of references to assist in setting policy aboard their vessels. Between DOD security manuals, DISA directives, and big Navy/CNO directives, ship commanders may receive instructions that leave too much room for interpretation or are unreasonably restrictive. This can be witnessed when communicating with sailors' currently assigned to ships and listening to how their CO is handling PEDs on those individual units. The disparity is a significant indicator that, at the ship level, the Navy is not ready to implement anything that looks like a BYOD policy.

In the submarine force, the range of limitations has swung wildly from allowing devices in certain portions of the boat to collecting and locking up all forms of PEDs. More recent policy has clearly outlined what is and is not allowed (Figure 3); however, as manufacturers increase device capabilities, simple devices like e-readers or mp3 players may have the ability to record data.[5] This may leave sailors on a short underway or even a whole deployment without access to readers, mp3 players, personal computers, etc. These were two different interpretations of the same policy. Commanders have tried to meet in the middle ground, restricting PEDs with certain capabilities from going into sections of their boat with FOUO or classified material. Photographs of reactor compartments and recordings of conversations create a risk of data being released regarding naval sub construction and layout or even capabilities and location. All of this is classified at a minimum secret level and may move into the top secret, compartmented access realm.

[5] "Portable Electronic Device Policy," 2013, http://kitsap.navylifepnw.com/modules/media/?do=download&id=f08e4efe-a340-4cf6-8d80-c6c96e3f7e2e, 10 (enclosure 2).

Figure 3. A table pulled from a submarine force instruction showing what devices and capabilities are permitted inside the boat



Device Capability	Examples	Additional Guidance
Cellular Devices	cell phones, smart phones, tablets w/cellular capability	-Use onboard prohibited. - May be brought and stored onboard while underway or in other than homeport. -Cellular capability should be disabled while onboard if possible.
Dedicated Audio Recording Devices	Digital or Analog Audio Recorders	-With exception of command owned devices, use onboard prohibited. -May be brought and stored onboard while underway or in other than homeport.
Dedicated Video Recording Devices	Single use (film) cameras, video cameras, digital cameras	- With exception of command owned devices, use onboard prohibited. -May be brought and stored onboard while underway or in other than homeport.
Wireless RF Device	2.4 or 5 GHz WiFi, Bluetooth, 802.11A/B /G/N, RF based wireless peripherals.	-OU-PED only; Use of P-PED WiFi onboard prohibited. -Devices with RF capabilities must have hardware disabled.
Wired Networking devices	Ethernet or USB	-Must not have wireless capabilities. -P-PEDs not authorized to connect to any OU-PED or ship's information system. -Wiring must not be run inside a ship's cable-run
Tablets	Galaxy, IPad, Kindle Fire, PanDigital, Blackberry Playbook, Google ANDROID	-Wireless capability must be disabled while onboard. -use of video or audio recording capabilities prohibited onboard.
Portable Gaming systems	XBOX, PlayStation, PSP, Nintendo Gameboy,	-use of wireless controllers and remotes are authorized -WiFi capability must be disabled
Digital Picture Frames		-Motion Sensor capable versions unauthorized -Audio Recording capable unauthorized
Portable Computers	Laptops, Netbooks, Notebooks	-Wireless capability must be disabled while onboard. -use of video or audio recording capabilities prohibited onboard.

Source: "Portable Electronic Device Policy," 2013, http://kitsap.navylifepnw.com/modules/media/?do=download&id=f08e4efe-a340-4cf6-8d80-c6c96e3f7e2e, Enclosure 2, 10.

In the surface fleet, the inconsistency is just as present but documentation is less specific. This is not the commander's fault but is the result of trying to allow reasonable use while restricting the potential for a security violation. Some ships have set up use areas, much like smoking areas, which are topside and safe for sailors to go check messages, respond to family or take a break on their device. Other commanders have a zero use policy, much like what was described for submarines above. Still others have written documents that are more traditional, just stating which spaces are not authorized for use of mobile devices. These spaces are usually radio, combat, some engineering spaces, or any other place where electronic classified information may be generated. This is a smart, liberal policy but does little for actual implementation. Sailors moving about the skin of the ship are eventually going to enter a location by mistake where they should not have their device. While this is a reportable security incident, the usual response is to leave the space quietly.

If commanders had the ability to disable those portions of a smart phone or tablet that caused concern they could work with the boat's security manager to establish a specific policy for which devices are allowed and how they must be configured. These devices would still not be allowed in DOD prohibited areas, but the devices would be set to nullify a security threat even before going inside a ship. An application that scans an input source and shuts down the camera, microphone, wireless data transfer methods, and any other write method, would provide a sense of security and control. Sailors running the application could easily show the device to leadership, who could spot check that the application is running and which hardware is disabled. If a security violation happens, it can be noted in the required report that the application was running and which pieces of hardware had been disabled. The days of writing loose policy that is poorly implemented or overly restrictive and that is burdensome on the sailor and leadership could be reduced significantly.

3. The DOD Security Concern

The Department of Defense wants to move towards utilizing devices that people already own for email, calendar, contacts, and limited access to encrypted data. There is no full solution at this time and the result has been enterprise procured devices that must be accounted for, distributed, and collected from users. In addition to creating another device for users to carry, these devices are generally a year or more behind current technology and are strapped to older hardware or operating systems and those vulnerabilities. DISA is developing software solutions to access classified content on government procured devices and hopes to move this out to personally procured devices in the future.[6]

For now, however, there is nothing in place for all of the mobile devices moving in and out of official, secure facilities that would limit the hardware from recording potentially harmful data. This is where a lockdown application has the potential to help manage the DODs movement towards a BYOD policy, give commanders the flexibility to control the hardware at the root level, and allow users to still have access to their devices in a more limited capacity while at work.

D. A RECENT EXAMPLE OF A SECURITY INCIDENT

In the last half of 2015, questions were being asked aboard USS *Alexandria* if unauthorized photos had been taken of sailors and spaces. The investigation eventually turned towards a specific sailor, MM1 Saucier, when his phone was found in a trash canister off the ship. After being questioned by the Federal Bureau of Investigation (FBI), someone attempted to destroy other devices before discarding them in the woods near Saucier's grandmother's house. Retrieval of the data showed that photos had been taken of "Alexandria's control panels, reactor compartment, and a monitor showing the sub's exact location at the time of the photo."[7]

The articles do not state where the ship was operating or the nature of the underway period. What is important to note is that, had this device been locked down with an application that disabled photographic and audio recording capability at the root level, this user would have been much less likely to have had the opportunity to commit this act. The real goal of locking down a device is to provide proof that PEDs can have a place on a warship and can be controlled by the command as desired. However, in the

[6] Kim Rice, "DOD Mobility, Presentation," Defense Systems Information Agency, June 17, 2015, http://www.disa.mil/~/media/Files/DISA/News/Conference/2015/Secure_MobilityRice.ashx.

[7] "Sailor Faces Charges after Photos of Navy Attack Sub Found on Cellphone," *Fox News*, August 3, 2015, accessed February 14, 2016, http://www.foxnews.com/us/2015/08/02/sailor-faces-charges-possessing-photos-navy-attack-sub-on-cellphone.html; David Larter, "Sailor Faces Charges for Submarine Photos on Cellphone," *USA Today*, August 1, 2015, http://www.usatoday.com/story/news/nation/2015/08/01/sailor-faces-charges-submarine-photos-cellphone/31005689.

immediate future it also provides a method to simply reduce the propensity to accidentally or intentionally commit violations of security practices. Reducing this will increase the security of the ship and help limit the loss of ship-specific information.

E. METHODS AND ORGANIZATION

The following methods are used to evaluate the ability to address security concerns and implement an application that could help address them:

- An examination of current Navy practices and policies with regards personal electronic devices (PEDs).
- An analysis of programming packages for Android that provided flexibility and access to root level permissions.
- Study of QR code uses and libraries, real world applicability, and utility in changing device settings using the camera as input.
- Study of NFC devices and on-device receivers, real world applicability, and utility in changing device settings with the NFC receiver as input.
- Investigation of which aspects of a phone capabilities cause the greatest concern for Navy and industry leadership with regard to security vulnerabilities associated with PEDs.

Below is a list of thesis chapters and a summary of content:

- Chapter II—Review of current literature for commercial mobile devices, and comparison of benefits to bring your own device (BYOD) policy changes. Discussion of how policy would allow and where BYOD could be effective if it is possible to implement.
- Chapter III—Technical aspects of intended research are examined including Android architecture. Each input/output (IO) method is examined individually so that appropriate functions may be incorporated in the development of the application. The benefits of QR and NFC input are discussed and analyzed with respect to ease of implementation.
- Chapter IV—Details of application development and implementation of desired functions and controls. Code samples for lock down methods settings changes using NFC are demonstrated. Testing and demonstration of the application and its implementation.
- Chapter V—Future work including BYOD implementation in a larger environment with emphasis on Navy-wide use. Where BYOD can be implemented in a beneficial manner, and in which cases will it still cannot be and why. Defining the potential for application development that will minimize or assist in elimination of security incidents related with mobile devices.

II. LITERATURE REVIEW

The overall aim of research conducted is directed at developing a security application for use on mobile devices used by sailors throughout the Navy. Our approach to literature review is a two-part process. First, we attempt to gain as much knowledge as possible in overall policy and instruction currently in place throughout the United States government with respect to security and mobile devices, applications, and networking. Second, after gaining insight in to the policies and instructions that would ultimately govern the use of a mobile security application, we shift our focus to Android based literature as we commence development. Routine reviews of both areas are conducted to ensure that our research employed the most current and up to date information and tactics. This chapter discusses current policies and instructions, while Chapter III will address the Android-specific technical considerations.

As mentioned above we direct the bulk of our early research efforts at thoroughly examining current and proposed policies and instructions governing technology. More specifically, we examine policy on how mobile devices are authorized for use, where they are authorized for use, and how our application development could allow greater use of personal electronic devices (PED) throughout the Navy enterprise. We take the approach of identifying each instruction and or policy and make a concerted effort to discuss how the contents of each relate to mobile application development. The intent is to provide insight into how development efforts and designs, as well as resulting products are affected by each policy or instruction.

A. DEPARTMENT OF DEFENSE

The Department of Defense (DOD) offers the best high-level scope of instruction and regulation with respect to mobile device and application development. We find that the DOD instructions governing mobile devices address agency procured devices and policies for their security, with little guidance governing personal devices. While the DOD does address restrictions for personal mobile devices through bans on use and connectivity, there is little other than plans for future work with respect to personal electronic device integration in the workforce. This naïve notion reflects the DOD's narrow concern with agency procured devices that access, use, store, or manage DOD information, while ensuring that personal devices do not touch (directly or indirectly) department information or systems. We agree with most DOD literature regarding personal mobile devices when considering the effects of unsecured access to Department of Defense networks by personal mobile devices. For this reason we feel that personal mobile device security is an essential, and relatively unaddressed area of concern within the DOD. By addressing each applicable information technology instruction in the DOD, we identify areas for improvement or recommend inclusion of personal mobile devices and how our security application could positively impact their security.

1. Department of Defense Commercial Mobile Device Implementation Plan

The DOD *Commercial Mobile Device (CMD) Implementation Plan* addresses a number of issues with respect to use of mobile devices throughout the department. First and foremost, the plan identifies the growing end user dependence on mobile devices and the importance of maximizing the availability of mobile devices.[8] The plan also identifies the Defense Information Systems Agency (DISA) as the DOD mobility program manager, and directs it to provide "secure classified and unclassified mobile communication capabilities to the DOD on a global basis."[9] As a direct result of this plan's directive to create a mobile application store (MAS), DISA has created and currently manages the DOD MAS through user portal access[10] (Figure 4). Another benefit of the plan is the creation of component MASs that provide tailored applications for services and are fully integrated and supported by the DOD MAS.[11]

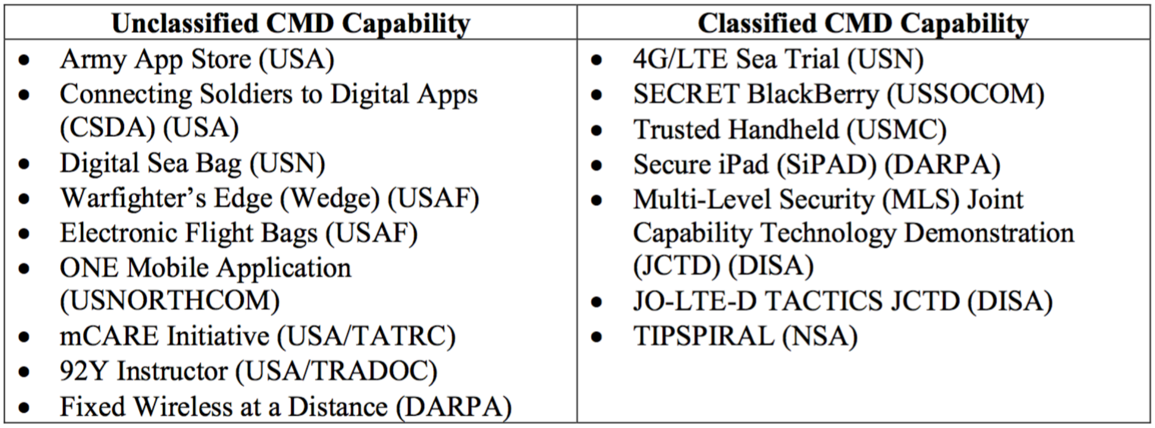
[8] Department of Defense, Chief Information Officer, *Commercial Mobile Device Implementation Plan* (Washington, DC: Department of Defense, 2013), http://archive.defense.gov/news/DoDCMDImplementationPlan.pdf, 3–4.

[9] Ibid., 3–4.

[10] Defense Information Systems Agency, "DOD Mobility Applications," accessed June 17, 2015, http://www.disa.mil/Enterprise-Services/Mobility/Apps.

[11] Department of Defense, *Commercial Mobile Device Implementation Plan*, 7.

Figure 4. Component mobility pilots



Unclassified CMD Capability	Classified CMD Capability
• Army App Store (USA) • Connecting Soldiers to Digital Apps (CSDA) (USA) • Digital Sea Bag (USN) • Warfighter's Edge (Wedge) (USAF) • Electronic Flight Bags (USAF) • ONE Mobile Application (USNORTHCOM) • mCARE Initiative (USA/TATRC) • 92Y Instructor (USA/TRADOC) • Fixed Wireless at a Distance (DARPA)	• 4G/LTE Sea Trial (USN) • SECRET BlackBerry (USSOCOM) • Trusted Handheld (USMC) • Secure iPad (SiPAD) (DARPA) • Multi-Level Security (MLS) Joint Capability Technology Demonstration (JCTD) (DISA) • JO-LTE-D TACTICS JCTD (DISA) • TIPSPIRAL (NSA)

List of component and service pilot programs identified in the Department of Defense, *Mobile Device Implementation Plan* (Washington, DC: Department of Defense, 2013), 7.

The aim of our application development is the submittal and approval of the app to Digital Sea Bag for use throughout the fleet. Furthermore, given the potential for securing mobile devices via our application, submittal to DISA's MAS for approval and ultimate use would assist in advancing mobile implementation throughout the DOD enterprise. We further address the specifics of the development life cycle and how the government's requirements for application development effect that life cycle in the DISA literature review.

2. Department of Defense Interoperability of Information Technology, Including National Security Systems Instruction 8330.01

The Department of Defense chief information officer seeks to improve interoperability of IT enterprise wide and thoroughly covers duties and responsibilities of each level of the Department of Defense in this instruction. To greater effect, DODI8330.01 provides a means for high level analysis of interoperability by "establishing a capability-focused, architecture-based approach for interoperability analysis."[12] DODI 8330.01 directs the director of DISA to "Aid the DOD Components

[12] Department of Defense, *Interoperability of Information Technology (IT), Including National Security Systems (NSS)* (DOD Instruction 8330.01) (Washington, DC: Department of Defense, 2014), https://acc.dau.mil/adl/en-US/706841/file/77077/DoD%20-%20Instruction,%20DoDI%208330.01,%20Interoperability%20of%20IT%20and%20NSS,%2021%20May%202014.pdf, Enclosure 2, 11.

with developmental IT interoperability testing to deliver solutions, reduce duplication of effort, and enhance IT interoperability".[13]

We find that our efforts at mobile application development on a widely used and well-known operating system, namely Android, assist in providing joint interoperability by utilizing technology that is already in the hands of DOD personnel. Expanding application development to other operating systems such as iOS only furthers this effort.

3. Department of Defense Cybersecurity Instruction 85000.01

Applicable across the DOD enterprise, with the exception of DOD SCI special access programs, DODI 8500.01 provides overarching insight into the mission, roles, and responsibilities in DOD information systems. Of special note is the department wide transition from using the term "Information Assurance" (IA) to the term "Cybersecurity."[14] Moreover, DOD8500.01 cancels its earlier version titled DOD Information Assurance. This revision has driven the U.S. Navy CIO to expand the scope of instructional coverage from IA defined areas of concern, namely information and information systems, to cybersecurity defined areas of "computers, electronic communications systems, electronic communications services, wire communication, and electronic communication, including information contained therein."[15] We feel that personal mobile devices reasonably fall within that scope or definition.

DOD 8500.01 addresses the issue of mobile device security with respect to DOD procured devices:

> DOD Components will ensure new computer assets (e.g., tablet, smartphone, personal digital assistant, mobile phone) procured to support DOD will include a technical performance measurement (TPM) version

[13] Ibid.

[14] Department of Defense, Chief Information Officer [DOD CIO], *Cybersecurity* (DOD Instruction 8500.01) (Washington, DC: Department of Defense, 2014), http://www.dtic.mil/whs/directives/corres/pdf/850001_2014.pdf.

[15] Department of Navy, Chief Information Officer, "DOD Instructions Lead to Change in Cybersecurity Term," Department of Navy Chief Information Officer, August 25, 2014,http://www.doncio.navy.mil/ContentView.aspx?id=5431.

> 1.2 or higher where required by DISA STIGs and where such technology is available.[16]

This is the sole reference to mobile device security however and it does not address personal devices used by DOD personnel. Given that this instruction replaces the DOD Information Assurance instruction (of the same number, 8500.01), and is written and designed in an attempt to bring the DOD up to speed with current technology, we find that it does not go far enough in realizing the true scope of personal mobile device security. An example is on page 43 of DOD 8500.01: "All IT that receives, processes, stores, displays, or transmits DOD information will be acquired, configured, operated, maintained, and disposed of consistent with applicable DOD cybersecurity policies, standards, and architectures."[17] The matter of DOD procured devices is addressed with respect to cybersecurity, but nothing is said about personal mobile devices. Given that 64 percent of American adults currently have smartphones, and 90 percent and higher of those individuals use their smart devices for Internet, video, voice and text,[18] the principle DOD instruction on cybersecurity should address security on these devices in a DOD environment. In an attempt to better equip DOD and Navy personnel to utilize their devices while at the same time secure them, we feel that applications like ours are essential and required to maintain enterprise wide security.

4. Department of Defense Risk Management Framework for DOD Information Technology (DODI 8510.01)

DODI 8510.01 provides "an integrated enterprise-wide decision structure for cybersecurity risk management."[19] As stated in previous sections of this chapter, any reasonable individual who has accessed the Internet via their mobile device would assume that personal mobile device security would fall into the definition of cybersecurity. This instruction lacks any direct reference to mobile devices, personal or

[16] DOD CIO, *Cybersecurity*.

[17] Ibid.

[18] Smith, *U.S. Smartphone Use in 2015*.

[19] Department of Defense, Chief Information Officer [DOD CIO], *Risk Management Framework (RMF) for DOD Information Technology (IT)* (DOD Instruction 8510.01) (Washington, DC: Department of Defense, 2014).

enterprise procured, with respect to risk management. What this instruction does provide is an authority in the following phrase: “The cybersecurity requirements for DOD information technologies will be managed through the RMF consistent with the principals established in National Institute of Standards and Technology (NIST) Special Publication (SP) 800–37.”[20] Further discussion on NIST is reserved for that section of the literature review, but worth noting is the intra-agency collaboration (DOD specifically) found within NIST guidance. We have found that the majority of DOD’s directives either directly or indirectly refer to NIST guidance based on the authorities found herein.

5. DOD Chief Information Officer Memo of 17 February 2012, Optimizing Use of Employee Information Technology Devices and Other IT to Achieve Efficiencies

Initially, and by title, this memo from the DODCIO showed promise by directing optimization of the department through the use of employee information technology. Upon further investigation however, this memo is concerned with optimizing the use of issued technology by eliminating redundancies in issued IT equipment. Solutions offered include offering kiosks for group in cases where individual employees do not require routine and regular use.[21] This instruction helps to emphasize what we address major faults in enterprise solutions by discussing acquisitions’ deliberate and burdensome processes, cost and budget issues, and redundant technologies.[22] There is no mention of the use of personal mobile devices in this memo. We add this memo to this literature review to demonstrate how seemingly applicable documentation within the DOD was often found to contain no guidance for personal mobile devices, and point out further how memos like these convolute a security professional’s ability to find guidance and make decisions for protection of systems.

[20] Ibid.

[21] Department of Defense, Chief Information Officer, *Optimizing Use of Employee Information Technology (IT) Devices and Other IT to Achieve Efficiencies* [memorandum] (Washington, DC: Department of Defense 2012).

[22] Ibid.

B. DEPARTMENT OF THE NAVY

Of all the departments' and agencies' instructional guidance, the Department of the Navy (DON) has the least technical coverage of mobile devices. Most importantly however is the overall lack of DON guidance on securing mobile device technology. Today more than ever in the past, sailors and marines utilize modern technology in their daily lives, including at-work and in-work spaces. One only needs to be underway on a U.S. Navy vessel for a day to realize how integrated mobile technology is in our sailors' lives. Without direct guidance from the DON Chief Information Officer (DONCIO) with up-to-date technology coverage, the future of mobile security will be limited at best.

1. Department of the Navy Enterprise Mobility 2008

DON *Enterprise Mobility 2008* shapes the need for mobility in the phrase "The end state capability to realize this vision will utilize 'smart' devices in the field…."[23] As an end state, this document delivers the vision for utilizing commercially available mobile devices. Recognizing the need for wireless solutions for sailors and marines, the document states, "The DON looks to a variety of commercially available wireless products to meet much of its enterprise mobility needs."[24] These goals are however limited to the scope of enterprise solutions, whereby mobility is encapsulated by the delivery mechanism of agency procured devices. The limited scope of the DONCIO vision is completely captured in this section of DONCIO *Enterprise Mobility 2008*.

> As technology convergence drives more power and functionality into smaller and smaller devices, such as smart phones, they become increasingly important in delivering enterprise mobility. Using commercial wireless products also enables standardization and interoperability across the Enterprise.

First, the desire for enterprise solutions limits the Navy to acquisitions requirements and budgetary constraints, similar to procuring any other part or component in the military. This results in significant lag time between available technology and the

[23] Department of Defense, Chief Information Officer, *Enterprise Mobility 2008* (Washington, DC: Department of the Navy, 2008).

[24] Ibid.

ability of the Navy to use it placing the department in a position where technology is outdated before it is acquired. Second, the aim of standardization of devices places the Navy in a position of choosing one device, or set of devices over another. This adds processes by inserting a procurement cycle, whereby testing, contracting, and budgets again take precedent over technology advancement and utilizing those devices that are already in most sailors' possession.

We seek to build a security application that begins to meet the vision of this document, while avoiding the burden of enterprise solutions with standardized devices. By developing an application to address security issues on mobile devices, technology continues to evolve without the Navy having to choose or invest device by device. Rather, the Navy invests in keeping this and potentially other applications up to date and as functional on as many devices as possible. The choice and cost of devices is then put in the hands of the sailor, and outside of current DON and DOD procurement processes.

2. Department of Navy (Plan for Optimizing Use of Employee Information Technology Devices and Other IT to Achieve Efficiencies

This memo issued by the DONCIO is the Navy's response to DODCIO's request for employee optimization of IT resources. As in the DOD issued memo, this memo focuses primarily on enterprise IT equipment such as government issued phones, laptops and tablets. By using enterprise solutions for the Navy's IT needs, whereby the government procures the requisite devices, the Navy is put in the position of having to not only buy, but also track and optimize the use of said devices. The level of effort required in such an endeavor is captured in the following phrase: "The DON has deployed tools that enable commands to track zero-use devices, minute optimization, air card costs, and roaming costs down to the individual level." Any reasonable individual could safely assume that several man hours would be spent in tracking and managing government issued IT. While we do not propose that our security application will solve all the efficiency issues designated in this or the associated DOD memo, putting mobility costs in the hands of the user will offer greater optimization over bureaucratic agency solutions. This is achieved through use of a personal mobile device with Navy issued applications, including our security application, whereby the user determines the when, where, and how, including which carrier they utilize, the amount of data they use, and pay for such services themselves.

3. DON Security Guidance for Personal Electronic Devices (DON CIO Message DTG: 202041Z AUG 07) and Subsequent Amplifications

This message contains DON guidance for the use of personal electronic devices (PED), one of the early terms used to describe mobile devices in the last half of the 2010. While still in use today, the term personal electronic devices usually applies to government procured and issued devices, as opposed to personally owned devices.

This message does detail some very important requirements for accessing DOD networks and more specifically DON email and associated accounts. As with many of the sources and guidance we have encountered in our literature review, this instruction is focused mainly with PED's that are procured through the government. Furthermore, this message provides what would be useful guidance for personally procured devices (bring your own device [BYOD]), such as "all PEDS must be capable of supporting digital signature and encryption (Secure/Multipurpose Internet Mail Extensions (S/MIME)) functionality."[25] This example S/MIME guidance does make its way into more recent documentation, especially throughout the NIST literature, but has not been updated to the specifications found in said NIST guidance.

Given that the date of this message places it in 2007, one could safely assume that the Navy at the time of writing was not yet fully aware of the relative technology explosion that has put personal electronic devices in the hands of many of its sailors. Just two percent of American cell phone subscribers owned a smartphone in 2005, and according to *Business Insider*, in 2007 global smartphone ownership was only three percent.[26] Interestingly enough, there has not been amplification or additional guidance to this instruction that should include today's technology and how it should access

[25] Department of the Navy, Chief Information Officer [DON CIO], *DON Security Guidance for Personal Electronic Devices* (202041Z AUG 07) (Washington, DC: Department of the Navy, 2007), http://www.doncio.navy.mil/uploads/0128BAA54339.pdf.

[26] Cathy De Rosa et al., *Perceptions of Libraries, 2010, Context and Community* (Dublin OH: Online Computer Library Center, 2010); John Heggestuen, "One in Every 5 People in the World Own a Smartphone, One in Every 17 Own a Tablet" *Business Insider*, December 15, 2013, http://www.businessinsider.com/smartphone-and-tablet-penetration-2013-10.

government networks. As it stands, this message series has amplification in 2009, stating, "PEDS operating without an associated smart card reader shall be disconnected on 31 December 2009."[27] This provides a framework for personnel to use their PED to access government networks, and given the somewhat recent procurement of mobile device CAC card readers by the DOD, there is relevance in such guidance.[28] Worth noting is that this access does come at an expense, where readers can cost anywhere between $99 and $369 making agency wide procurement or personal purchase costly.[29]

According to this naval message, "all PED interconnections must be made using a designated accrediting authority (DAA) approved device through either a physical connection or a secured Bluetooth communications link."[30] This is especially important today as multiple mobile games now support linking through Bluetooth, and anyone who has been underway on a U.S. navy vessel can find sailors utilizing Bluetooth linking of devices. This is an important area to address in our mobile application, as data could theoretically be shared between two Bluetooth devices, one inside, and one outside a secure area up to 100m apart.[31] While future implementations of our application could include specific Bluetooth connection block types (i.e., device to device), at the current stage of development we simply lockout all access to Bluetooth (discussed further in Chapter IV).

C. NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY

From its founding in 1901, NIST has been the source of information and standards in the United States. For example, the more technical, original aspects for

[27] Department of the Navy, Chief Information Officer, *Amplification Guidance for Purchase and Installation of Personal Electronic Device Smart Card Readers* (281919Z JAN 09), (Washington, DC: Department of the Navy, 2007).

[28] "Tactivo Order from US Department of Defense," Precise Biometrics, December 17, 2014, http://precisebiometrics.com/news/2014/12/tactivo-order-us-department-defense/.

[29] Michael J. Danberry, "MilitaryCAC's Information on Using Your CAC with Your Mobile Device including AKO Email," 'January 13, 2016, https://militarycac.com/mobile.htm.

[30] DON CIO, *DON Security Guidance for Personal Electronic Devices*.

[31] Joshua Wright, "Dispelling Common Bluetooth Misconceptions," Security Laboratory: Wireless Security, September 19, 2007, http://www.sans.edu/research/security-laboratory/article/bluetooth.

electricity were managed and promulgated through NIST.[32] Today, many departments rely on the standards set forth by NIST, including the DOD. The vast majority of DOD and DON publications regarding technology either reference NIST standards or take standards directly from their publications. With 399 publications on computer security, and 712 documents related to information technology, the NIST provides a wealth of guidance on practices, policies, and procedures related to technology. Important to the reader is that NIST not only uses its own staff for research but also draws on industry experience from the Defense Advanced Research Projects Agency (DARPA), Office of the Director of National Intelligence (ODNI), Department of Defense (DOD), Committee on National Security Systems (CNSS), Department of Homeland Security (DHS), Department of Justice (DOJ), and various corporations with extensive industry participation in NIST's guidance delivery undertakings. For the purpose of this research, we focus on NIST standards relate to mobility, mobile applications, mobile security, and mobile device utilization, and recognize it as the foremost authority on mobile device guidance based on breadth and depth of coverage in the area of concern.

The NIST typically publishes special bulletins related to overarching topics and guidance in broad categories of interest. It also issues bulletins that update or amplify information for an associated special publication. This is essential to our research as the majority of the special publications we accessed had amplifying guidance set forth in a follow on bulletin.

1. NIST SP 800–53 Revision 4 Security and Privacy Controls for Federal Information Systems and Organizations (SP 800–53r4)

This publication ties the authorities and requirements established at the highest levels of government through the Federal Information Security Management Act (FISMA) and the Federal Information Processing Standards (FIPS) to specific security requirements for use within each agency of the government. More specifically this instruction provides "guidelines for selecting and specifying security controls for organizations and information systems supporting the executive agencies of the federal

[32] National Institute of Standards and Technology, "The Story of NIST," National Institute of Standards and Technology, last updated February 24, 2014, http://www.nist.gov/timeline.cfm.

government to meet the requirements of FIPS Publication 200, *Minimum Security Requirements for Federal Information and Information Systems*."[33] The security controls discussed in this document provide the base level of our first problem statement whereby we seek to determine what we need to do via our application to secure a mobile device.

As discussed in the introduction to the NIST, collaboration is an important part of the guidance they provide. This is clear in the list of SP-800-53r4 contributors that includes "NIST, the Department of Defense, the Intelligence Community, and the Committee on National Security Systems as part of the Joint Task Force, an interagency partnership formed in 2009."[34] NIST here includes private sector experts from top industry partners such as Mitre, Booz Allen Hamilton and Johns Hopkins APL. The DOD collaborators include its Chief Information Officer and his technical directors, including cybersecurity. The intelligence community provides its inputs from its director (DNI), via the national intelligence chief information officer, and the intelligence community security risk manager. We point this out and list its principal partners because based on our research, SP 800–53r4 has the broadest and highest ranking group of technical security professionals of any of the documents we reviewed. Based on this collective expertise, we recognize this document as the baseline document for security within information systems (IS) including mobile devices.

We find our first standard definition of a mobile device in this instruction. While we later discuss a more thorough definition in a later NIST publication, this one certainly gave us the highest level technical attempt at defining mobile devices. SP 800–53r4 gives a four plus definition whereby devices are broadly identified across four areas, plus additional possible quantifiers. The first trait identified by this publication is that the device "has a small form factor such that it can easily be carried by a single individual"[35] Simple and strait forward, by this trait laptops, smart phones, tablets, apply. Second, the device "is designed to operate without a physical connection (e.g., wirelessly transmit or

[33] Department of Commerce, and National Institute of Standards and Technology, *Security and Privacy Controls for Federal Information Systems and Organizations* (SP800-53r4), rev. 4 (Gaithersburg, MD: U.S. Department of Commerce, and National Institute of Standards and Technology, 2013).

[34] Ibid.

[35] Ibid.

receive information)"[36] Here is the notion of mobility. While essential to the definition of a mobile device, we find this property too vague, and better defined in other NIST publications discussed later in this chapter. Third, the device must "possess local, non-removable or removable data storage." This portion of the definition is generally applied throughout the majority of the literature we found that references or attempts to define mobile devices. The notion of mobile data storage created serious security concerns for the Navy resulting in complete bans on USB thumb drives and other removable media on its networks.[37] Technically speaking introduction of malware through removable media is a valid and effective attack vector for malicious behavior in business, government and personal computing. Fourth, the device "includes a self-contained power source."[38] This part of the definition does not play a major role in the overall mobile device specification other than to point out that the device has a life cycle where it can operate independent of power. This differentiates a mobile device from the USB drive or CD, which require power provisioning. The additional features covered in the "four plus" definition include some of the most important mobile features especially from a security standpoint. Among those is the onboard sensors and/or built in local and remote data synchronization.[39] Finally, SP 800–53r4 provides its clearest description of mobile devices through its examples that include "smart phones, tablets, and E-readers."[40]

The purpose of this instruction is to provide a security control framework to allow federal organizations to effectively and as completely as possible implement security controls over technology in the workplace. To accomplish this SP 800–53r4 provides control recommendations and selection criteria for those recommendations. SP 800–53r4 then breaks those categories into eighteen families (Figure 5).[41] In the broadest sense

[36] Ibid.

[37] Commander Navy Cyber Forces, *Commander's Cyber Security and Information Assurance Handbook* (COMNAVCYBERFORINST 5239.2A), rev 2 (Washington DC: Department of the Navy, 2013).

[38] Ibid.

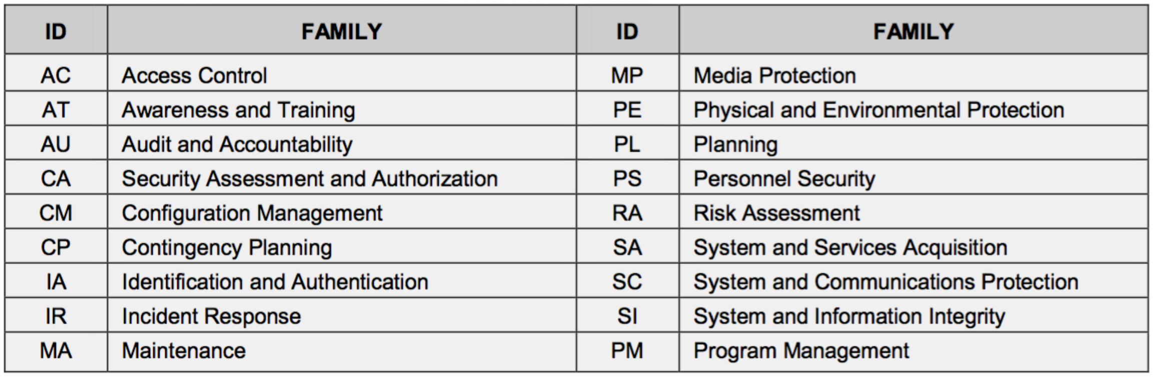
[39] Ibid.

[40] Ibid.

[41] Ibid.

possible we address those controls that NIST associates with mobile devices and attempt to explain how those controls govern the use of mobile devices and how our application might provide the recommended control. With respect to the applicable mobile device controls, we found the following five controls addressing mobile devices.

Figure 5. Security control identifiers and family names



ID	FAMILY	ID	FAMILY
AC	Access Control	MP	Media Protection
AT	Awareness and Training	PE	Physical and Environmental Protection
AU	Audit and Accountability	PL	Planning
CA	Security Assessment and Authorization	PS	Personnel Security
CM	Configuration Management	RA	Risk Assessment
CP	Contingency Planning	SA	System and Services Acquisition
IA	Identification and Authentication	SC	System and Communications Protection
IR	Incident Response	SI	System and Information Integrity
MA	Maintenance	PM	Program Management

Source: National Institute of Standards and Technology*, Security and Privacy Controls for Federal Information Systems and Organizations* (SP 800–53r4), 2013, http://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-53r4.pdf, 9.

AC 19 provides direct guidance with respect to mobile devices and access control. The most definitive statement in this control section states that the organization "establishes usage restrictions, configuration requirements, connection requirements, and implementation guidance for organization-controlled mobile devices."[42] While directed at organization controlled devices (read enterprise), we feel that this control applies to personally owned mobile devices, and that our application could be used to provide and enforce usage restrictions, and configuration management. The control amplification section gives further guidance on what implementation of this control is acceptable as follows: "configuration management, device identification and authentication, implementation of mandatory protective software."[43] Our application could be used to identify users in a database implementation that associates users to devices (discussed further in Chapter V), and could be classified as protective software. Another key feature

[42] Ibid.

[43] Ibid.

that we address in our application's current configuration is "disabling unnecessary hardware (e.g., wireless)."[44]

This control method further recommends, "unclassified mobile devices and the information stored on those devices are subject to random reviews and inspections and if classified information is found, the incident handling policy is followed." We feel that by offering full lockdown on our application and implementing that at access points to areas where secure information exists, we significantly reduce the need for intrusive searches of personal property by restricting the ability for the information to get on the device. Furthermore, with the current display that shows the state of connectivity related features on the device, security checks could be accomplished by viewing the display to ensure proper controls for the applicable space are implemented, and if a user is found in violation of the required setting, said inspections of their device could then be completed.

(1) AC-20 Use of External Information Systems

The control statement in Access Control (AC) 20 gives guidance on "the use of non-organizationally owned information systems, system components, or devices to process, store, or transmit organizational information."[45] AC-20 defines personal mobile devices as independent information systems. In the "Additional Control Recommendations" AC-20 lists four restrictions related to the use of non-organizational information systems (including personal devices), three of which are directly applicable to our research. The first recommendation relates directly to our application by calling for the "implementation of organization-approved security controls prior to authorizing such connections."[46] Our application could be one of the Navy security controls that users are required to install under the authority of this instruction. The second restriction recommendation states that an organization can "limit access to certain types of information, services, or applications." We interpret this as authority to lock out access to features on mobile devices, exactly how our application locks out Wi-Fi, Bluetooth,

[44] Ibid.

[45] Ibid.

[46] Ibid.

Camera, Data etc. The third restriction does not directly apply to the current implementation of our application as it discusses virtualization and storage on servers.[47] The fourth recommended restriction provides the authority for users to utilize their device in the workplace by requiring the them to "agree to terms and conditions for usage."[48]

(2) CA-9 Internal System Connections

Security Access and Assessment (CA) provides the authority to approve or deny network access by clearly defining the type of connection, either for individual connections or connection groups. The control implemented by this instruction calls for the documentation of each connection (or group of connections), the interface characteristics, and the security requirements for use of that connection or interface, and the information travelling by means of that connection or interface.[49] Our application directly supports this control by specifically restricting access to connection interfaces. An example of this control in use could include identifying smart phones as a connection group, and require that Wi-Fi be secured while onboard the ship.

One additional security control provided by CA-9 is the notion that "smart phones be configured with a specific baseline configuration."[50] This baseline configuration could require that our application be installed, and furthermore the application could be modified to verify a baseline configuration upon installation.

(3) SC-42 Sensor Capability and Data

System and Communication Protection (SC) 42 speaks directly to the use of mobile devices, and covers access to sensors that process, transmit or are activated by environmental data. An example provided by SC-42 is GPS. The control provides for the restriction or complete prohibition of access to such features, and points to the threat of covert activation and collection by an adversary.[51] The example provided by SC-42 is

[47] Ibid.

[48] Ibid.

[49] Ibid.

[50] Ibid.

[51] Ibid.

something that we tried to convey to the crew of the ships we were on whereby an adversary "remotely activates the GPS function on a mobile device and gains the ability to track the specific movements of an individual" or unit.[52]

(4) MP-7 Media Use

Media Protection (MP) 7 provides for restriction or prohibition of devices with information storage capability. Two important recommendations are made in this control, specifically organizational restriction on the type of device authorized for use such as personally owned devices, and disabling or removing capability to write to the device.[53] Anyone who has been underway on a U.S. Navy vessel has seen implementation of the first restriction, as mobile devices are normally restricted from use in any space with classified information. This restriction results in many of the issues discussed in Chapter I, whereby verification and reporting of personnel with mobile devices in secure spaces is a heavy burden and lacking universal application on the rule. The second recommendation is more aptly served by our application and approach, as in future implementations of our application, we could prohibit device access to onboard storage. Upon activation of lockdown via NFC swipe, the device would be locked out of write capability. This and other work is discussed further in Chapter V.

2. Special Publication 800–124 Revision 1 Guidelines for Managing the Security of Mobile Devices in the Enterprise

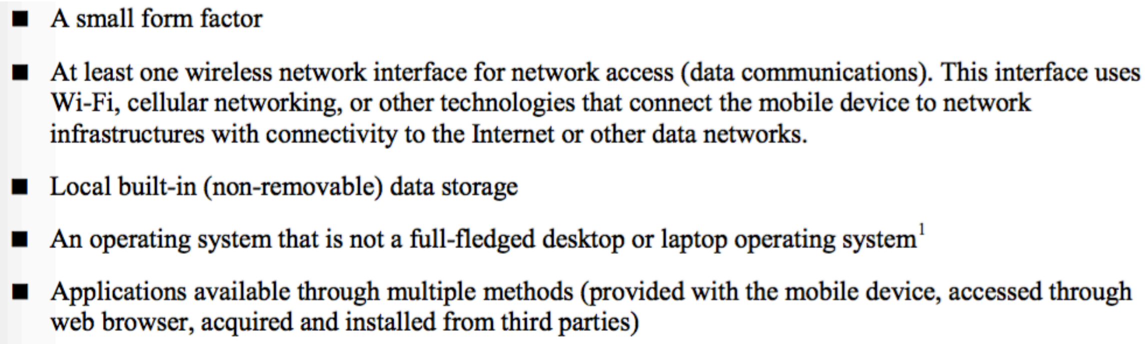
As we seek to provide a security application for the Navy to utilize in secure mobile devices, we sought out a definition for mobile device that would encapsulate the vast majority of the devices we seek to secure, while at the same time drawing a distinct line between a mobile device and the more traditional laptop or desktop computer. For the purposes of our research we find that NIST SP 800–124 provides a concrete definition of "mobile device" providing the proper scope of interoperability and definitive characteristics. Furthermore, many of the characteristics listed therein are the very characteristics we seek to lock out as they are potentially significant security threats.

[52] Ibid.

[53] Ibid.

While basic considerations will be covered in this chapter, the specifics of why these features are considered a threat to security will be covered in Chapter III, technical aspects of research, and where explanations are given instructionally. SP 800–124 defines the mobile device in the following list (Figure 6):

Figure 6. NIST SP 800–124 mobile device characteristics



- A small form factor
- At least one wireless network interface for network access (data communications). This interface uses Wi-Fi, cellular networking, or other technologies that connect the mobile device to network infrastructures with connectivity to the Internet or other data networks.
- Local built-in (non-removable) data storage
- An operating system that is not a full-fledged desktop or laptop operating system[1]
- Applications available through multiple methods (provided with the mobile device, accessed through web browser, acquired and installed from third parties)

Source: National Institute of Standards and Technology*, Security and Privacy Controls*.
A comprehensive and efficient list of characteristics defining mobile devices.

NIST SP 800–124 goes further with features it describes as “optional.” SP 800–124r1 does identify these features as “particularly important in terms of security risk.”[54] In the case of our research, and especially as far as the Navy is concerned, we feel that these additional features should be included as primary in any future instructional guidance that governs mobile devices. The additional features that NIST SP 800–124 lists are (1) network services, (2) presence of camera and or video recording component, (3) microphone, and (4) storage. As it pertains to our research, the SP800-124 additional features are what we would in fact find to be vital features, as they are the very features we seek to lock out with our application. While some features discussed below are not present in every single smart phone or tablet, including them as universal features for lockdown will provide the greatest breadth of coverage in preventing access to mobile device features.

[54] Murugiah Souppaya, and Karen Kent, *Guidelines for Managing the Security of Mobile Devices in the Enterprise* (SP800-124r1) rev 1 (Washington, DC: National Institute of Standards and Technology, 2013).

(1) Network Services

SP 800–124r1 lists Bluetooth and NFC in the optional characteristics. As far as our research is concerned, both of these features offer the potential to pass information from one device to another and should be controlled and lockable. Another feature listed in the optional network services section is one or more wireless network interfaces for voice communication such as cellular.[55] We find that these features are also essential for our definition of mobile device, and further identify these features as ones that we seek to lock out with our application. Voice communication and other form of communication with or attempts to communicate via cell phone towers are features that could pose a threat to security. Finally, Global Positioning Systems that enable location services considered optional in SP 800–124r1 are for the purposes of our research considered a security threat.

(2) Digital Camera or Video Recording Device

Considering that anyone purchasing a phone or tablet would have reasonable difficulty finding a device without a camera onboard, we find that this feature is absolutely essential to the definition of a mobile device (with the exception of some e-readers). A search for tablet on Google and Amazon does not allow the search including an option for "no camera," rather only allows specification of the resolution. It is easy to understand why the military would want to limit, if not completely lock out, access to a camera given the level of security and classification of the material in the areas where a mobile device might be present. For this reason, and the aforementioned prevalence of cameras in mobile devices, we feel that further instructional guidance should include the feature as one of the definitive features rather than optional.

[55] Ibid.

(3) Microphone

In similar fashion to the camera feature, microphones are extensively prevalent in the mobile device market. They are fundamental to the operation of the communication aspect of the devices. The use of applications such as Facetime, Skype and Google+ allows users to communicate via video using their onboard cameras and microphones. Where one is present, one can generally assume the other does as well. The microphone is different, however, in that its use in a secure space can be more covert since the device could be inside an individual’s pocket, purse, or backpack, and be actively recording audio. Considering that a clandestine recording like this occurring in a sensitive or highly classified space could be detrimental to the confidentiality of the material discussed, we consider this feature vital to the definition of a mobile device.

(4) Storage

Non-removal data storage is included in the baseline definition of NIST SP 800–124r1, but support for removable media and support for the device itself functioning as removable storage for another computing device is not. Rather, the last two features are listed as optional. The Navy currently does not allow the use of removable media due to the security threat it poses on its networks.[56] Extensive policies are in place throughout the fleet and shore commands prohibiting users from inserting a personal storage device into any official navy computing device.[57] For this purpose we do not address this feature in our application, rather we attempt to prevent the use of any feature that creates data (such as the image file created using the camera, or the voice file created using the microphone) thereby preventing storage of potential threat creating application data.

(5) Synchronization of Local Data with a Different Location

The vast majority of personal mobile devices used at sea on naval vessels are used for reading, studying, gaming, listening to music, and movie viewing. This is not an all-inclusive list of potential activities, but certainly captures the bulk of the uses for mobile

[56] Commander Navy Cyber Forces, *Commander’s Cyber Security.* ’

[57] ’Ibid.

devices on Navy vessels. Given that every activity listed has local data that is stored on the device, and requires communication to access existing libraries stored in the cloud and new purchases, we feel that this is another essential base line characteristic for an accurate mobile device definition.

One of the initial assumptions in our research is the notion that everyone's device is unsecured. NIST SP 800–124r1 supports this and states, "Many mobile devices, particularly those that are personally owned (bring your own device, BYOD), are not necessarily trustworthy."[58] Furthermore, NIST SP 800–124r1 clarifies, "There is frequent jail-breaking and rooting of mobile devices, which means that the built-in restrictions on security, operating system use, etc., have been bypassed."[59] The notion of rooted devices would frighten any reasonable individual in a security role where mobile devices are present, as security controls are easily removed. We therefore recommend that rooted devices be prohibited as part of policy restrictions for use on DOD and DON networks.

Again, this instruction provides commentary on device security as it states, "Organizations should assume that all mobile devices are untrusted unless the organization has properly secured them and monitors their security continuously while in use with enterprise applications or data." We believe this requirement is partially accomplished by securing the device with our application and providing an on screen presentation of device feature status. Additionally, we propose future work discussed in Chapter V for continuous monitoring solutions that could enable manager notification in the event that our mobile application was altered or in cases where attempts were made to deactivate its functionality.

The common approach used by most secure areas in the Navy is to prohibit device presence inside the area. NIST SP 800–124r1 states exactly that in discussing options for bring your own device (BYOD): "One option is to restrict or prohibit use of BYOD devices, thus favoring organization-issued devices."[60] This is the very heart of our

[58] Souppaya and Kent, *Guidelines for Managing the Security of Mobile Devices*.

[59] Ibid.

[60] Ibid.

research as we seek to open the organization up to the use of personal devices by securing features that pose risks to security through our application. By controlling a device's access to features such as the camera, data transfer, Bluetooth and WiFi, the device is placed in a controlled state that prevents the user from transmitting harmful code or documenting classified material. This is an option discussed in NIST SP 800–124r1. The obvious difference is that we seek to use an individual's personal device as opposed to a device issues by an organization. As Souppaya, and Kent describe, "Another effective technique is to fully secure each organization-issued mobile device; this gets the mobile device in as trusted a state as possible, and deviations from this secure state can be monitored and addressed."[61] We believe we can put a personal mobile device in a secure state as discussed early by controlling its features that pose a security threat. The final option offered by NISP SP 800 124r1 is as follows: "There are also technical solutions for achieving degrees of trust in BYOD devices, such as running the organization's software in a secure, isolated sandbox/secure container on the mobile device, or using device integrity scanning applications."[62] This is an option we discuss in Chapter IV, as partitioning a device and initiating a separate managed administration profile by NFC is possible. In fact, this offers the benefit of the Navy authorizing specific applications for use within the managed profile.

In the following statement, NIST SP 800–124r1 points at a lack of central management and the resulting requirement for manual and individual management: "If there is not a centralized management solution, or certain mobile devices cannot use it, then mobile devices have to be managed individually and manually."[63] This is a common belief among mobile security instructions. We disagree on both points made in this phrase, as we seek to minimize central management by allowing the application to control access to features that represent security threats, and at the same time eliminate the need for individual or manual management, by having the application installed on all devices that will be brought on the ship when an individual checks in to the command.

[61] Ibid.

[62] Ibid.

[63] Ibid.

In its observation that "it may not be possible to manage the security of the device when it is not physically present within the enterprise." NIST SP 800–124r1 misses a control opportunity by ignoring the presence of a mobile application that resides on the device, and is initiated upon accessing the enterprise. The technical aspects of how this works in our application are covered more thoroughly in Chapter IV, but for sake of discussion, an application that launches and closes upon entry and exit to DOD areas respectively offers a wealth of upgradability and configurability by its very nature. With the advent of near field communication (NFC) and the ability to program tags that launch the application, updates can be pushed to the application upon initiation.

One of the most important parts of the publication is the general policy guidance with respect security on mobile devices within the enterprise. Our application meets several of these recommendations directly and eliminates the need for those remaining. For the sake of clarity, in Table 1, we list each recommendation and how we answer the recommendation with our application, through future work potential, or list as not applicable due to application control of the feature.

Table 1. NIST SP 800–124r1 recommendations correlated to the features of our application

NIST SP 800–124r1 Recommendation	**Research and Comments Related to NIST Requirement**
GENERAL	
Restrict user and application access to hardware.	Our application takes control of device hardware and locks out access to it while the application is running
Restrict user and application access to native OS services, such as the built-in web browser, email client, calendaring, contacts, application installation services, etc.	Future work could lockdown each application individually. Web browsing and email are both controllable via device policy administration. All remaining applications could be controlled via broadcast receivers or direct API lockout.
Manage wireless network interfaces (Wi-Fi, Bluetooth, etc.)	Our application directly restricts access to WiFi and Bluetooth

NIST SP 800–124r1 Recommendation	Research and Comments Related to NIST Requirement
Automatically monitor, detect, and report when policy violations occur, such as changes from the approved security configuration baseline, and automatically take action when possible and appropriate	Alarms and notifications to security managers could be built into future implementations of our application.
Limit or prevent access to enterprise services based on the mobile device's operating system version (including whether the device has been rooted/jailbroken), vendor/brand, model, or mobile device management software client version (if applicable). Note that this information may be spoofable.	Enterprise services such as access to command networks is currently controlled by IT infrastructures Navy wide. Future access could be contingent upon verification of application installation and operation, but this is not an issue on a personal device until this access is made available by DOD.
DATA COMMUNICATION AND STORAGE	
Strongly encrypt data communications between the mobile device and the organization	Encryption of communications is an optional feature for future development discussed in Chapter V, but Android provides default onboard encryption.
Strongly encrypt stored data on both built-in storage and removable media storage.	Restricting access to onboard and removable storage while in lock down is an area of future research identified in Chapter V, but Android provides default onboard encryption and it is an optional feature for removable media.
Wipe the device (to scrub its stored data) before reissuing it to another user, retiring the device, etc.	Not applicable to BYOD.
Remotely wipe the device (to scrub its stored data) if it is suspected that the device has been lost, stolen, or otherwise fallen into untrusted hands and is at risk of having its data recovered by an untrusted party.	This is an option in the Android OS and the recommendation should be to have it turned on for users. This could be addressed with training and device security policy.
A device often can also be configured to wipe itself after a certain number of incorrect authentication attempts.	See comment above.
USER AND DEVICE AUTHENTICATION	

NIST SP 800–124r1 Recommendation	Research and Comments Related to NIST Requirement
Require a device password/passcode and/or other authentication (e.g., token-based authentication, network-based device authentication, domain authentication) before accessing the organization's resources.	Access to command networks is controlled by local instruction and according to Navy guidance. These accesses currently require login and authentication.
If device account lockout is enabled or the device password/passcode is forgotten, an administrator can reset this remotely to restore access to the device.	As our application is locked and unlocked via NFC, in the case where neither happens as designed, a master NFC tag can be encoded to always lock/unlock all app features.
Have the device automatically lock itself after it is idle for a period (e.g., 5 minutes).	Typically OS specific, but it is a feature and best practice for Android. This could be addressed with training and device security policy.
Under the direction of an administrator, remotely lock the device if it is suspected that the device has been left in an unlocked state in an unsecured location.	This remote feature is available in Android and provides users a way to lock their lost/ misplaced device.
APPLICATIONS	
Restrict which app stores may be used	When in locked status, or on an authorized command network, screening of websites or stores accessible would be controlled locally according to Navy instructions on approved wed sites. Additionally, policy and training should recommend/prohibit access to third party application stores.
Restrict which applications may be installed through whitelisting or blacklisting	Not addressed by our application.
Restrict the permissions (e.g., camera access, location access) assigned to each application.	Our application fully locks out access to network features, thereby removing each application's access to that feature.
Install, update, and remove applications. Safeguard the mechanisms used to perform these actions. Keep a current inventory of all applications installed on each device.	By locking out access to features at the device level, other applications are unable to access them, thereby removing the risk of malicious application actions (such as

NIST SP 800–124r1 Recommendation	Research and Comments Related to NIST Requirement
	opening a microphone in the background).
Restrict the use of operating system and application synchronization services (e.g., local device synchronization, remote synchronization services and websites).	Not addressed in the current implementation of our application but possible in future work.
Verify digital signatures on applications to ensure that only applications from trusted entities are installed on the device and that code has not been modified.	Not addressed by our application and not required. However, as capabilities open as discussed in Chapter V, this would be a consideration for authentication.
Distribute the organization's applications from a dedicated mobile application store.	DISA currently maintains at DOD application store, as does the Navy through Navy Seabag. Our aim is to have a future version of our application included and available through these stores.

Adapted from National Institute for Standards and Technology, *Guidelines for Managing the Security of Mobile Devices in the Enterprise* (SP800-124), rev 1 (Washington, DC: NIST, 2013).

3. The Information Technology Laboratory at the National Institute of Standards and Technology ITL Bulletin for July 2013

This bulletin is used to implement many of the government procured device security recommendations for use of personal devices. In fact this bulletin directs organizations to utilize the controls and policies implemented for organizational devices when implementing or authorizing personal mobile device use.[64] This ITL recommends the very approach we have taken to securing mobile devices. Specifically, we utilize as many of the security controls for organizationally owned devices as possible with respect to personal mobile device security recommendations.

4. Special Publication 800–163 Vetting the Security of Mobile Applications

NIST SP 800–163 provides the general guidance for the process and requirements for approval of mobile applications. This document was the framework for design

[64] Elizabeth Lennon, "ITL Issues and Guidelines for Managing the Security of Mobile Devices," *ITL Bulletin*, July 2013. http://csrc.nist.gov/publications/nistbul/itlbul2013_07.pdf.

considerations while building our applications. Many of the topics covered here are also considered by Google and other application stores with the addition of more restrictive security controls.

The first category in the requirements for approval is that the application have authorized functionality. By this, the instruction further describes that the application must work as described, and error conditions must be clearly identified and notify the user. The second category requires that the application not contain any unauthorized functionality such as data exfiltration and malware. We accomplish this by only allowing the application to store the state of the device as locked or unlocked as discussed in Chapter IV. The third category evaluates the permissions granted to the application, and recommends limiting them to only those necessary for operation. Our application only operates with those permissions required to lockdown the device and does not contain any unnecessary permissions. The fourth category requires that the application protect sensitive data. The application we developed does not currently handle user information and future implementations would only handle usernames associated to device IDs. The fifth category addresses code dependencies. These dependencies include the need for and use of libraries. This drove the design of our application as discussed in Chapters III and IV, and at the current time our application's only dependency relies on knowing the state of the device. The sixth category tackles testing app updates. Updates to our application are discussed further in Chapter IV, but thorough testing would be part of any update.

Another vital part of SP 800–163 for our development was the testing approaches section. The approval process for vetting applications is to include four basic steps: "(1) correctness testing, (2) analysis of the app's source code or binary code, (3) the use of static or dynamic analysis, and (4) manual or automatic testing of the app."[65] We specifically wrote and re-wrote the code used in our application to ensure source code correctness, and statically and dynamically tested it as discussed in Chapter IV.

[65] Steve Quirolgico et al., *Vetting the Security of Mobile Applications* (SP-800-163) (Washington, DC: National Institute of Standards and Technology, 2015).

D. DEFENSE INFORMATION SYSTEMS AGENCY

Drawing its roots from the technology utilized in World War II and the need for management of such technology DISA now serves as the principle authority for certification and regulation of information assurance and security in the Department of Defense.[66] DISA operates in and supports the DOD and the Whitehouse communications office, with combatant command field offices embedded in all COCOMs and joint operations centers (JOCs).[67]

DISA served as a principle source of information with respect to our research in the use of personal mobile devices in the DOD workplace. By design, DISA is the forward leaning technology arm of the DOD, as evidenced through its Mobile Device Management (MDM) program for enterprise procured devices used in classified and unclassified capacities. The program itself functions as a DOD certified interface that "provides the commercial mobile device (CMD) and user level controls necessary to enforce security policies within and for the use of the mobile device."[68] By creating a managed profile on the device, the MDM is able to "institute policy, security, and permissions that define the functions enabled on the mobile device."[69] DISA elaborates on the specific management and security functions of the profile capabilities as follows: "(MDM) Supports malware detection, over-the-air (OTA) electronic software distribution of applications, remote data-wipe capabilities, remote device configuration management, and asset/property management capabilities that protect against key and data compromise."[70]

One of the keys to DISA's MDM functionality relies on the use of enterprise procured devices. DISA identifies the equipment it manages as government furnished

[66] Defense Systems Information Agency, "Our History: The Beginnings, the Creation of DCA, 1947–1960, Post World War II/Cold War," Defense Systems Information Agency, accessed February 1, 2016, http://www.disa.mil/About/Our-History.

[67] Defense Information Systems Agency, "Our Organization Structure," Defense Systems Information Agency, accessed February 1, 2016, http://www.disa.mil/About/Our-Organization-Structure.

[68] Defense Information Systems Agency, "DOD Mobility Unclassified Capability," accessed February 1, 2016, http://www.disa.mil/Enterprise-Services/Mobility/DMUC.

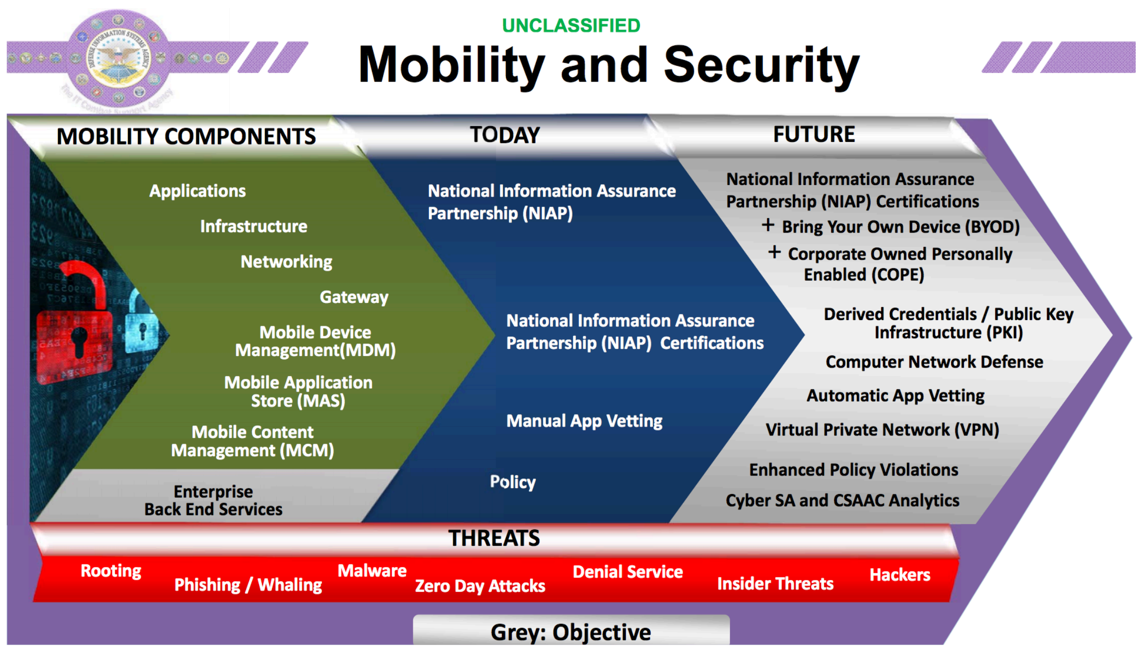
[69] Ibid.

[70] Ibid.

equipment (GFE), and stipulates that any devices that will operate on its MDM must be GFE. DISA does go further by providing regular updated authorized device lists that improve the functionality of its MDM program. The list of approved devices correlates strongly with technology currently available to the consumer market, with offerings such as iPhone 6 and 6 Plus, and Samsung Galaxy S6 and S6 Edge.[71] With respect to BYOD, DISA does identify the use of personally owned devices within the DOD as an area of future growth as seen in Figure 7.

[71] Department of Defense, Mobility Program Management Office, *Memorandum for DOD Mobility Supported Devices* (Fort Meade, MD: DOD Mobility Program Management Office, 2015).

Figure 7. DISA brief on mobility and security



UNCLASSIFIED

Mobility and Security

MOBILITY COMPONENTS

Applications

Infrastructure

Networking

Gateway

Mobile Device Management(MDM)

Mobile Application Store (MAS)

Mobile Content Management (MCM)

Enterprise Back End Services

TODAY

National Information Assurance Partnership (NIAP)

National Information Assurance Partnership (NIAP) Certifications

Manual App Vetting

Policy

FUTURE

National Information Assurance Partnership (NIAP) Certifications

+ Bring Your Own Device (BYOD)

+ Corporate Owned Personally Enabled (COPE)

Derived Credentials / Public Key Infrastructure (PKI)

Computer Network Defense

Automatic App Vetting

Virtual Private Network (VPN)

Enhanced Policy Violations

Cyber SA and CSAAC Analytics

THREATS

Rooting

Phishing / Whaling

Malware

Zero Day Attacks

Denial Service

Insider Threats

Hackers

Grey: Objective

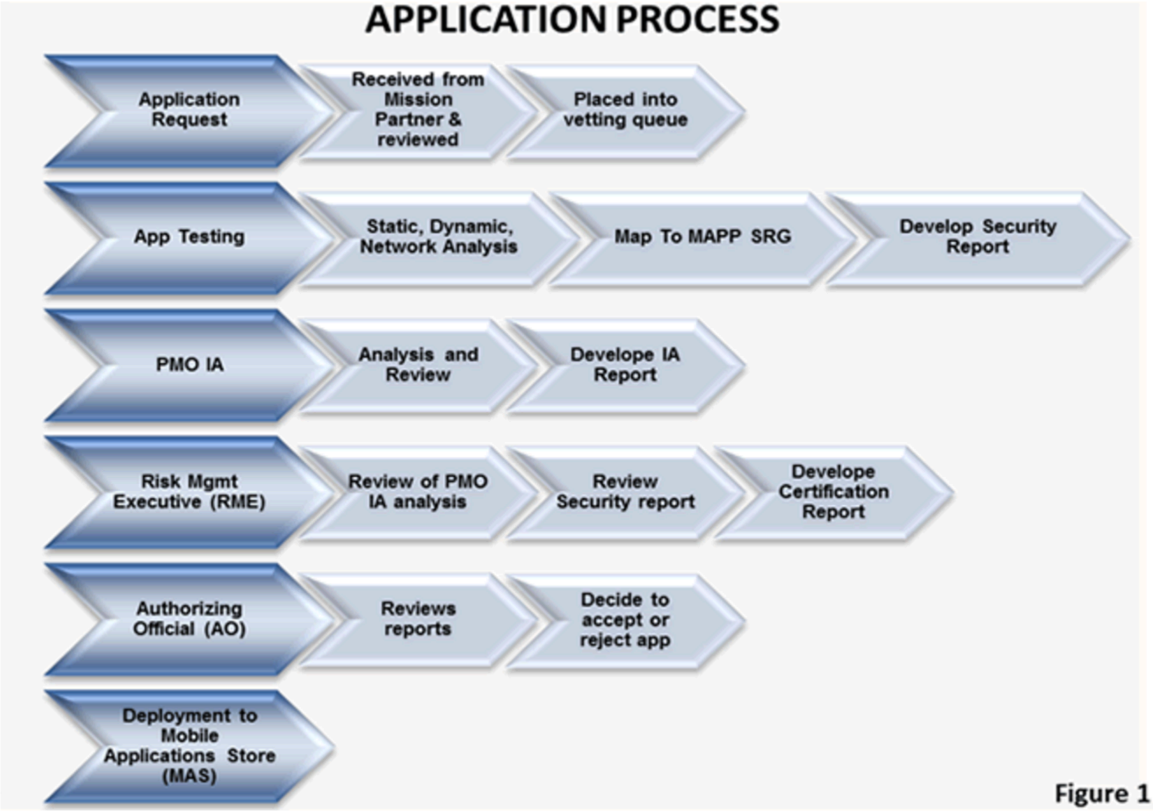
Source: Kim Rice at the Mobile Project Management Office identified BYOD as an area of future growth and research in her 17 June 2015 DOD mobility brief. Kim Rice, "DOD Mobility, Presentation," Defense Systems Information Agency, June 17, 2015, http://www.disa.mil/~/media/Files/DISA/News/Conference/2015/ Secure_MobilityRice.ashx.

1. DOD Mobility Applications

As our research pertains to the development of a mobile application to aid in the security of personal mobile devices, we sought out DISA guidance for such development. We found that while DISA is forward leaning in the MDM sector, the application development area appears at best burdensome. Based on the current model offered by DISA, the process from inception to implementation is six steps with multiple testing and report development phases (Figure 8).[72] While the need for extensive testing and vetting of security flaws in applications designed for use by the DOD presents a significant challenge, the overall process as currently implemented is time and build prohibitive based on Figure 8.

[72] Defense Information Systems Agency, "DOD Mobility Applications."

Figure 8. DISA mobility application development process


APPLICATION PROCESS
Application Request
Received from Mission Partner & reviewed
Placed into vetting queue
App Testing
Static, Dynamic, Network Analysis
Map To MAPP SRG
Develop Security Report
PMO IA
Analysis and Review
Develope IA Report
Risk Mgmt Executive (RME)
Review of PMO IA analysis
Review Security report
Develope Certification Report
Authorizing Official (AO)
Reviews reports
Decide to accept or reject app
Deployment to Mobile Applications Store (MAS)
Figure 1

DISA's mobile application development process showing six stages with multiple testing and report generating phases.

DISA is currently working on its own IDE that would allow for significant streamlining of application development in its MDM program. Their proof of concept model is currently called the Application Development Platform, and seeks to provide drag and drop UI with cross platform and OS interoperability.[73] Based on DISA's description this IDE would be similar to Google's Android Studio, with one major exception: DISA seeks to provide an HTML hybrid code development suite.[74] This will be a massive undertaking considering the scope of mobile operating systems and lack of interoperability between them. An example is the Java based Android suite and the C# based iOS environment that requires coding in largely different formats and according to different design principles. Coding in either Java or C#/SWIFT allows direct access to

[73] Ibid.

[74] Ibid.

root function controls such as Device Policy Administrator. We explored coding environments such as Ionic that enable HTML coding with implementation across both Android and iOS operating systems, but were cumbersome in accessing root features. We found that additional training was required and after multiple attempts did not provide as much functionality as writing the app to be OS specific. Using an HTML hybrid IDE similar to Ionic could provide a roadblock to rapid development of complex, root level access applications in DISA's application environment.

E. SECURITY TECHNICAL IMPLEMENTATION GUIDES

Security technical implementation guides (STIGs) are developed by the Defense Information Systems Agency (DISA). For the purposes of our research, we evaluated several STIGs in the areas of mobility and wireless security. We found extensive guidance on the use and associated security of commercial mobile devices (CMDs). As with other sources explored in our literature review, we found very little guidance on the use of personal mobile devices in the workplace. Furthermore, we reached out twice seeking further guidance and received no feedback. For the purposes of our research, we will evaluate STIG policy associated with CMDs and enterprise solutions, identifying areas where BYOD could be applicable.

STIGs offer guidance and findings associated with Information Assurance. The goal of the guidance is to provide "technical guidance to 'lock down' information systems/software that might otherwise be vulnerable to a malicious computer attack."[75] Each STIG is identified within a category by its associated STIG ID, and includes its severity code. These severity codes indicate the level of security compromise and are summarized for technical use by Josef Weiss in his article "STIG Alerts (by CAT)" as follows:[76]

[75] Defense Information Systems Agency, "Security Technical Implementation Guides (STIGs)," Defense Information Systems Agency, May 21, 2015, http://iase.disa.mil/stigs/Pages/index.aspx.

[76] Josef Weiss, "STIG Alerts (by CAT)," Tenable Network Security, December 18, 2014, https://www.tenable.com/sc-dashboards/stig-alerts-by-cat.

- CAT I—allows "primary security protections to be bypassed, allowing immediate access by unauthorized personnel or unauthorized assumption of super-user privileges."[77]
- CAT II—"potential to lead to unauthorized system access or activity."[78]
- CAT III—"recommendations that will improve IA posture."[79]

For the purposes of our research as a proof of concept in mobile security application development, we focus on the CAT 1 severity code issues first then address applicable CAT II severity code issues.

STIGS are grouped in security requirement guides (SRG) according to the genre to which they apply. We evaluate the two SRGs associated with mobility as they contain the widest breadth of applicable STIGs, and are the SRG against which the commands are evaluated during and information assurance accreditation, certification, and evaluation inspection. The majority of these STIG's are essential to this and any other research in the area of personal mobile device utilization within the DOD and the Navy. In order to understand how our research relates to each STIG, we introduce the STIG by its number, followed by the associated rule taken verbatim from the STIG. The recommended check and associated fix are also verbatim from the STIG. We then provide a general discussion of BYOD related issues and how our application development applies either in current form, future development potential, or as not applicable. Those STIG's not applicable to our research are labeled as such, with brief rule descriptions and reasoning for non-applicability.

1. Mobile Policy Security Requirements Guide Release: 2 Benchmark Date: 26 Jul 2013

Mobile Policy Security SRG covers policy, procedure and settings recommendations for agency wide use of mobile devices. "Mobile Policy Security Requirements" SRG contains 71 rules associated with mobile policy security requirements within the DOD. Of the 71 rules, nine are CAT I, 35 are CAT II, and 27 are

[77] Ibid.

[78] Ibid.

[79] Ibid.

CAT III. Numbers one through nine are rules with CAT I severity, numbers 10 through 45 are rules with CAT II severity.

(1) STIG ID: SRG-MPOL-015

Rule: "The organization must remove the wireless interface on computers with an embedded wireless system before the computer is used to transfer, receive, store, or process classified information."[80]

Recommended fix: "Remove computers with embedded wireless interfaces that cannot be removed from all classified use; these computers must not transfer, receive, store, or process classified information."[81]

Discussion: This STIG does not apply to mobility on the device end or discuss security requirements there-in.

(2) STIG ID: SRG-MPOL-017

Rule: "The organization must ensure all wireless systems connected to a DOD network (including associated peripheral devices, operating system, applications, network/PC connection methods, and services) are approved by the approval authority prior to installation and use for processing DOD information."[82]

Recommended fix: "Obtain DAA approval, documented by memo or site security plan, prior to wireless systems connected to a DOD network being installed or utilized."[83]

Discussion: DOD networks are not approved for use of non-agency procured mobile devices. Later STIG's specifically address this issue and identify what networks non-agency procured devices are authorized to access. Our application can assist in preventing the peripheral device (mobile device) from accessing the network however, by locking out WiFi access.

[80] Defense Information Systems Agency, *Mobile Policy Security Requirements Guide* (SRG-MPOL-015), July 13, 2013, https://www.stigviewer.com/stig/mobile_policy_security_requirements_guide/2013-07-03/finding/V-35935, ID: V-35935.

[81] Ibid.

[82] Ibid.

[83] Ibid.

(3) STIG ID: SRG-MPOL-020

Rule: "The organization must maintain a SIPRNet connection approval package with the Classified Connection Approval Office (CCAO) when connecting a Secure WLAN (SWLAN) to SIPRNet."[84]

Recommended fix: "Disable or remove the non-compliant SWLAN until the site has all required approvals for operation."[85]

Discussion: This specific STIG is not applicable to mobile devices specifically, but does address the approval needed for creating a SWLAN. Our application could assist in this effort by blocking access to WiFi in general from the device.

(4) STIG ID: SRG-MPOL-037

Rule: "The organization must have written policy or training material stating CMDs must not be used to receive, transmit, or process classified messages unless specifically approved by NSA for such purposes and NSA-approved transmission and storage methods are used."[86]

Recommended fix: "Develop and publish policy preventing CMDs from processing, sending, receiving, or storing classified data."[87]

Discussion: Commercial Mobile Devices are generally defined within DISA and these STIG's as agency procured and provisioned devices. In the spirit of trying to closely design policies and procedures related to agency procured and provisioned CMD's for the use of personally owned mobile devices, we find that this limitation on CMD's applies directly to BYOD. Furthermore our application can lock out access to

[84] Defense Information Systems Agency, *Mobile Policy Security Requirements Guide* (SRG-MPOL-020), January 24, 2013, https://www.stigviewer.com/stig/mobile_policy_security_requirements_guide/2013-01-24/finding/V-35938, ID: V-35938.

[85] Ibid

[86] Defense Information Systems Agency, *Mobile Policy Security Requirements Guide* (SRG-MPOL-037), July 3, 2013, www.stigviewer.com/stig/mobile_policy_security_requirements_guide/2013-07-03/finding/V-35955, ID: V-35955.

[87] Ibid.

WiFi and Bluetooth thereby completely eliminating the ability to receive or transmit classified messages.

(5) STIG ID: SRG-MPOL-040

Rule: “The organization must have a policy forbidding the use of wireless personal area network (PAN) devices, such as near-field communications (NFC), Bluetooth, and ZigBee, to send, receive, store, or process classified information. The check applies to Wireless USB (WUSB) devices…however, it does not apply to wireless email devices (BlackBerry, Windows Mobile, etc.)”[88]

Recommended fix: “Develop and publish a policy forbidding the use of wireless PAN devices for classified processing.”[89]

Discussion: Our application can lock out device access to NFC and Bluetooth with growth work potential for lock out of all other wireless PAN mediums, thereby eliminating the threat to sending or receiving classified information.

(6) STIG ID: SRG-MPOL-042

Rule: “The organization must have written policy or training material that states non-enterprise activated CMD are not permitted to connect to DOD networks.”[90] There is a significant risk of introducing malware on a DOD network if these types of devices are connected to a DOD network. Allowed “exception: the device can be connected to a DOD managed Internet-Gateway-only connected Wi-Fi access point (AP).”[91] The organization requires approval from the authorizing official for the connection of unclassified mobile devices to unclassified information systems.

[88] Defense Information Systems Agency, *Mobile Policy Security Requirements Guide* (SRG-MPOL-040), July 3, 2013, www.stigviewer.com/stig/mobile_policy_security_requirements_guide/2013-07-03/finding/V-35958, ID: V-35958.

[89] Ibid.

[90] Defense Information Systems Agency, *Mobile Policy Security Requirements Guide* (SRG-MPOL-042), July 3, 2013, www.stigviewer.com/stig/mobile_policy_security_requirements_guide/2013-07-03/finding/V-35960, ID: V-35960.

[91] Ibid.

Fix: “Develop and publish the policy or procedure preventing connection of CMDs and tablets classified as non-enterprise activated to DOD networks and users are trained on the requirement.”[92]

Discussion: Our application currently offers two states for the device, either locked down or unlocked. Growth potential exists such specifically authorized network connections could be authorized, discussed further in Chapter VI. The DOD managed Internet-Gateway-only network would provide access (when device is unlocked) for sailors to social media sites, online media and other materials. By offering the capability to lockout out the devices wireless accesses, the Navy could control access times and locations, for example limiting wireless access during the workday thereby minimizing device distractions.

(7) STIG ID: SRG-MPOL-052

Rule: “The organization must follow the incident handling policy if classified information is found on mobile devices.”[93]

Recommended fix: “Ensure the organization has defined an incident handling policy with specific actions to be implemented when classified information has been found on mobile devices… Follow all incident handling policy actions to be taken when classified information has been identified on mobile devices.”[94]

Discussion: Later STIG’s address what specifically has to happen when classified information is found on a mobile device, namely data wiping. Our application steps in front of this need by allowing the commander to set lock/unlock locations where the application delivers its lockdown state to the device. In the case of this STIG specifically, by locking down access to the camera in the current implementation, and access to data storage in future implementations, our application could provide a mechanism for eliminating the threat identified by this STIG.

[92] Ibid.

[93] Defense Information Systems Agency, *Mobile Policy Security Requirements Guide* (SRG-MPOL-052), July 3, 2013, www.stigviewer.com/stig/mobile_policy_security_requirements_guide/2013-07-03/finding/V-35970, ID: V-35970.

[94] Ibid.

(8) STIG ID: SRG-MPOL-058

Rule: “The organization must not use DOD-issued software certificates for Non-enterprise activated CMDs.”[95]

Recommended fix: “Publish the organization’s implementation guidance prohibiting the use of DOD-issued software certificates on non-enterprise activated CMDs.”[96]

Discussion: In its current form, and as stated above, our application functions in two states, either fully locked or unlocked. As discussed in Chapter VI, the growth potential for plug ins or increasing the number of states could eliminate the device’s ability to download or access DOD-issued software certificates.

(9) STIG ID: SRG-MPOL-069

Rule: “The organization must develop procedures for ensuring mobile operating systems, mobile applications, and mobile device management agents on managed mobile devices are updated within an organization defined period after the updates/patches are available.”[97]

Recommended fix: “Develop procedures to update mobile operating systems, mobile applications, and mobile device management agents on managed mobile devices within the organization defined period after the updates or patches are available.”[98]

Discussion: While this STIG addresses managed mobile devices, we feel that this requirement directly applies to our application and to the use of personally procured devices. Any reasonable individual understands that software patches are part of the framework of good cyber security practices. Outdates OS’s missing patches and updates

[95] Defense Information Systems Agency, *Mobile Policy Security Requirements Guide* (SRG MPOL-058), January 24, 2013, https://www.stigviewer.com/stig/mobile_policy_security_requirements_guide/2013-01-24/finding/V-35976, ID: V-35976.

[96] Ibid.

[97] Defense Information Systems Agency, *Mobile Policy Security Requirements Guide* (SRG-MPOL-069), July 3, 2013, www.stigviewer.com/stig/mobile_policy_security_requirements_guide/2013-07-03/finding/V-35987, ID: V-35987.

[98] Ibid.

are a prime target for malware. Our application could be grown to include a mechanism for checking the OS build number and deterministically locking out WiFi on DOD networks until the OS has been brought up to the most recent update. The specifics of exactly how this could be accomplished are covered in depth in Chapter VI, but it is worth mentioning here that this feature would be a more advanced feature requiring extensive development and testing, as well as a robust reference database of device hardware and software, requiring routine management and technical oversight.

(10) STIG ID: SRG-MPOL-001

Rule: "The organization must define the maximum number of consecutive, unsuccessful login attempts to CMDs are permitted."[99]

Fix: "Clearly define the maximum number of consecutive unsuccessful login attempts to the mobile device in its access control policy and/or security procedures."[100]

Discussion: Theft or loss of a CMD poses a significant security threat if the individual in possession of the device is given multiple attempts to login to it. Unlimited attempts to unlock the device only increase the threat of a brute force attack on the password space of the device. Lock out after a predetermined number of login attempts is an important feature of the agency procured device, or enterprise solution and we feel the same is true of a personal mobile device. For that reason, we feel that if an individual uses their personal device on a DOD Internet Gateway network, they should be subject to device lockout after a designated number of incorrect attempts. This will increase vigilance in personal device management, as well as deter to some extent the thefts that occur while underway on a Navy vessel. This requirement could be spelled out in a site security instruction and or built into our application.

[99] Defense Information Systems Agency, *Mobile Policy Security Requirements Guide* (SRG-MPOL-001), July 3, 2013, www.stigviewer.com/stig/mobile_policy_security_requirements_guide/2013-07-03/finding/V-35910, ID: V-35910.

[100] Ibid.

(11) STIG ID: SRG-MPOL-003

Rule: "The organization must make a risk-based determination for applications before they are accredited by the DAA prior to distribution or installation on a CMD."[101]

Fix: "Include a risk-based determination and DAA accreditation for applications prior to installation on a CMD in the CMD policy."[102]

Discussion: DISA Mobility User Corner provides the mechanism for conducting a risk based determination of an application. In fact, the risk based determination for applications is part of the application approval process required by DISA for use within the DOD.

(12) STIG ID: SRG-MPOL-005

Rule:

> The organization must monitor for unauthorized wireless connections to the information system at an organization defined time period…DOD components will ensure a Wireless Intrusion Detection System (WIDS) is implemented that allows for monitoring of WLAN activity and the detection of WLAN-related policy violations on all unclassified and classified DOD wired and wireless LANs.[103]

Fix: "Monitor for unauthorized wireless connections to the information system at an organization defined time period."[104]

Discussion: WIDS could provide the essential back end protection for the lockout feature in our application. Currently, attempts to access network features while in lock down are blocked by our application. The state of the device is stored while running and upon power cycling is restored. Upon installation of the application a registration of the

[101] Defense Information Systems Agency, *Mobile Policy Security Requirements Guide* (SRG-MPOL-003), July 3, 2013, www.stigviewer.com/stig/mobile_policy_security_requirements_guide/2013-07-03/finding/V-35912, ID: V-35912.

[102] Ibid.

[103] Defense Information Systems Agency, *Mobile Policy Security Requirements Guide* (SRG-MPOL-005), July 3, 2013, www.stigviewer.com/stig/mobile_policy_security_requirements_guide/2013-07-03/finding/V-35919, ID: V-35919.

[104] Ibid.

device user and associated MAC address could provide a means of identifying unregistered devices to the WIDS.

(13) STIG ID: SRG-MPOL-006

Rule: “The organization must define a time period for monitoring of unauthorized wireless connections to information systems, including scans for unauthorized wireless access points.”[105]

Fix: “Define the time period for monitoring of unauthorized wireless connections to information systems to include the time period for performing scans to identify unauthorized wireless access points.”[106]

Discussion: as stated in STIG SRG-MPOL-005, simple registration of device and associated identifying information could be gathered and stored following install of our application as discussed in Chapter IV. Furthermore, future work could include alarms and notifications that identifying unauthorized access to wireless connections.

(14) STIG ID: SRG-MPOL-007

Rule: “The organization must document and take appropriate action if an unauthorized wireless connection is discovered.”[107]

Fix: “Update documented procedures to document and take appropriate action if an unauthorized wireless connection is discovered.”[108]

Discussion: This rule can have two specifically associated scenarios. The first would be unauthorized attempt is made to connect to a wireless network with our application is installed but remained unlocked. In this case, and as discussed in Chapter VI, growth potential within our application could provide a full lockdown of the device.

[105] Defense Information Systems Agency, *Mobile Policy Security Requirements Guide* (SRG-MPOL-006), July 3, 2013, www.stigviewer.com/stig/mobile_policy_security_requirements_guide/2013-07-03/finding/V-35920, ID: V-35920.

[106] Ibid.

[107] Defense Information Systems Agency, *Mobile Policy Security Requirements Guide* (SRG-MPOL-007), July 3, 2013, www.stigviewer.com/stig/mobile_policy_security_requirements_guide/2013-07-03/finding/V-35921, ID: V-35921.

[108] Ibid.

This could be done via registered IP addresses where wireless connection with personal devices is unauthorized. The second scenario is a device that does not have our application installed, in which case the intrusion detection software would have to detect the attempted connection.

(15) STIG ID: SRG-MPOL-008

Rule: "The organization must define the appropriate action(s) to be taken if an unauthorized wireless connection is discovered…Wireless technologies include, but are not limited to, microwave, satellite, packet radio (UHF/VHF), Wi-Fi, and Bluetooth."[109]

STIG Fix Recommendation: "Define and document the appropriate action(s) to be taken when unauthorized wireless connections are discovered."[110]

Discussion: This is the policy piece of STIG SRG-MPOL-007 where defining and documenting appropriate actions to be taken in the event of unauthorized wireless connections are discovered. With respect to the implementation of our application, we would recommend that the application place the phone in complete lockdown if an attempt to access an unauthorized network was made while the device was unlocked. In the event that the device was locked, there is no capability for accessing a network.

(16) STIG ID: SRG-MPOL-009

Rule: "The organization must confine Wi-Fi and Bluetooth communications to organization-controlled boundaries…Ensure the organization has defined and established organization-controlled boundaries for the implementation of Wi-Fi and Bluetooth communications."[111]

[109] Defense Information Systems Agency, *Mobile Policy Security Requirements Guide* (SRG-MPOL-008), July 3, 2013, www.stigviewer.com/stig/mobile_policy_security_requirements_guide/2013-07-03/finding/V-35922, ID: V-35922.

[110] Ibid.

[111] Defense Information Systems Agency, *Mobile Policy Security Requirements Guide* (SRG-MPOL-009), January 24, 2013, www.stigviewer.com/stig/mobile_policy_security_requirements_guide/2013-01-24/finding/V-35928, ID: V-35928.

Fix: "Define and establish organization controlled boundaries for the implementation of the Wi-Fi and Bluetooth communications."[112]

Discussion: These boundaries will be defined by the commander and the site authorization to operate (ATO). With that in mind, our application can maintain device access to and from these boundaries. When outside an approved boundary, the application could be modified in future implementations to lock down access to specific Bluetooth or wireless networks.

(17) STIG ID: SRG-MPOL-010

Rule: "The organization must establish usage restrictions for wireless access…Implementing wireless computing and networking capabilities in accordance with the organization defined wireless policy, and allowing only authorized and qualified personnel to configure wireless services, greatly reduces vulnerabilities."[113]

Fix: "Establish a usage restrictions policy for wireless access within the organization's boundaries/enclave/area of responsibility."[114]

Discussion: Our security application targets this STIG directly, by controlling wireless access to networks. In current form, it eliminates all access, placing the device in a safer mode when in the vicinity of an unauthorized network or space. In future iterations, it could be modified to prevent access to specific networks, while allowing access to others, such as an Internet gateway network in a common space such as the mess decks or wardroom.

(18) STIG ID: SRG-MPOL-012

Rule: "The organization concept of operations (CONOPS) or site security plan must include guidance that signal amplification, antenna configuration, or other

[112] Ibid.

[113] Defense Information Systems Agency, *Mobile Policy Security Requirements Guide* (SRG-MPOL-005), October 10, 2012, www.stigviewer.com/stig/mobile_policy_security_requirements_guide/2012-10-10/finding/SRG-MPOL-005, ID: SRG-MPOL-005.

[114] Ibid.

techniques must not be modified in Bluetooth radios that could affect signal detection or interception."[115]

Fix: "Update CONOPS or site security plan to include Bluetooth radios must not be modified through signal amplification, antenna configuration, or other techniques that could affect signal detection or interception."[116]

Discussion: This STIG is not directly related to our research.

(19) STIG ID: SRG-MPOL-016

Rule: "The organization must establish implementation guidance for wireless access…Implementing wireless computing and networking capabilities in accordance with the organization defined wireless policy, and allowing only authorized and qualified personnel to configure wireless services, greatly reduces vulnerabilities."[117]

Fix: "Establish clear guidance for the implementation of wireless access within the organization's boundaries/enclave/area of responsibility."[118]

Discussion: Similar to STIG's SRG-MPOL-007 through SRG-MPOL-010 our application would ideally be managed and configured by command security manager and or information security officer. The current version does not enable variable settings, but in future implementations and as discussed in Chapter VI, these features could be added to allow the manager to configure device specific setting with respect to wireless access.

[115] Defense Information Systems Agency, *Mobile Policy Security Requirements Guide* (SRG-MPOL-012), July 3, 2013, www.stigviewer.com/stig/mobile_policy_security_requirements_guide/2013-07-03/finding/V-35930, V-35930.

[116] Ibid.

[117] Defense Information Systems Agency, *Mobile Policy Security Requirements Guide* (SRG-MPOL-016), July 3, 2013, www.stigviewer.com/stig/mobile_policy_security_requirements_guide/2013-07-03/finding/V-35934, ID: V-35934.

[118] Ibid.

(20) STIG ID: SRG-MPOL-024

Rule: "The organization must only procure and deploy WPA2-Enterprise certified WLAN equipment and software for wireless systems that connect directly to DOD networks."[119]

Fix: "Update all WLAN equipment and software to WPA2-Enterprise certified for wireless systems that connect directly to DOD networks."[120]

Discussion: As non-agency procured devices are not currently authorized to operate on DOD networks this STIG does not directly apply to our research. Future implementations of our application could however mandate that WPA-2 be the sole connection setting for personal mobile devices operating on DOD networks.

(21) STIG ID: SRG-MPOL-028

Rule: "The organization must authorize wireless access to the information system prior to connection."[121]

Fix: "Establish a wireless access control and security policy to define the administrative procedures and technical requirements to be met prior to being authorized to connect to an organization's information system(s)."[122]

Discussion: This STIG provides a perfect policy for requiring all individuals and their devices to be registered and have our application installed prior to permitting access to command networks. Doing so would further enable STIG SRG-MPOL-005 requirements for WIDS detection of unapproved connections to command networks.

[119] Defense Information Systems Agency, *Mobile Policy Security Requirements Guide* (SRG-MPOL-024), July 3, 2013, www.stigviewer.com/stig/mobile_policy_security_requirements_guide/2013-07-03/finding/V-35942, ID: V-35942.

[120] Ibid.

[121] Defense Information Systems Agency, *Mobile Policy Security Requirements Guide* (SRG-MPOL-028), January 24, 2013, www.stigviewer.com/stig/mobile_policy_security_requirements_guide/2013-01-24/finding/V-35946, ID: V-35946.

[122] Ibid.

(22) STIG ID: SRG-MPOL-032

Rule: "The organization must notify the Certified TEMPEST Technical Authority (CTTA) before a Secure WLAN (SWLAN) becomes operational and connected to the SIPRNet."[123]

Fix: "Confirm and document the local CTTA has been notified of the site's intent to install and operate a SWLAN."[124]

Discussion: This STIG is not directly related to our research or BYOD, but our application can assist in ensuring the TEMPEST certification remains intact by locking out network, mic and camera features when an individual swipes an NFC tag outside a secure space.

(23) STIG ID: SRG-MPOL-035

Rule: "The organization must ensure the network access control solution supports wireless clients and solutions if wireless networking is implemented."[125]

Fix: "Update the network access control solution to support all wireless clients and devices."[126]

Discussion: Currently, CANES allows wireless networking aboard ships and with future implementations this feature will only become more prevalent throughout the Navy. That said, our application could assist in such implementation by restricting access when needed, and allowing it when authorized. Furthermore, by allowing individuals to utilize their personal devices the DOD and Navy could realize a significant cost reduction.

[123] Defense Information Systems Agency, *Mobile Policy Security Requirements Guide* (SRG-MPOL-032), July 3, 2013, www.stigviewer.com/stig/mobile_policy_security_requirements_guide/2013-07-03/finding/V-35950, ID: V-35950.

[124] Ibid.

[125] Defense Information Systems Agency, *Mobile Policy Security Requirements Guide* (SRG-MPOL-035), July 3, 2013, www.stigviewer.com/stig/mobile_policy_security_requirements_guide/2013-07-03/finding/V-35953, ID: V-35953.

[126] Ibid.

(24) STIG ID: SRG-MPOL-038

Rule: “The organization must not permit operation of unclassified wireless devices in areas where classified information is electronically stored, processed, or transmitted unless operation is in accordance with DAA-approved CTTA restrictions at the site.”[127]

Fix: “Do not permit operation of wireless devices in areas where classified information is electronically stored, processed, or transmitted unless operation is in accordance with DAA-approved CTTA restrictions at the site.”[128]

Discussion: Our application and associated research aims directly at satisfying this requirement. By locking out networking features on a mobile device, we knock out the intentional or unintentional access to the threat vector of a mobile device in a classified setting.

(25) STIG ID: SRG-MPOL-043

Rule:

> The organization must not permit non-enterprise activated CMDs to process or store DOD sensitive information, including DOD email. There is a high risk of introducing malware and exfiltration of information if these types of devices store or process anything other than non-sensitive information.[129]

Fix: “Develop and publish the policy or procedure preventing the processing or storing of DOD sensitive information, including DOD email, by non-enterprise activated CMDs.”[130]

[127] Defense Information Systems Agency, *Mobile Policy Security Requirements Guide* (SRG-MPOL-038), July 3, 2013, www.stigviewer.com/stig/mobile_policy_security_requirements_guide/2013-07-03/finding/V-35956, ID: V-35956.

[128] Ibid.

[129] Defense Information Systems Agency, *Mobile Policy Security Requirements Guide* (SRG-MPOL-043), January 24, 2013, www.stigviewer.com/stig/mobile_policy_security_requirements_guide/2013-01-24/finding/V-35961, ID: V-35961.

[130] Ibid.

Discussion: When in a lockdown state, our application prevents access to any wireless network. With respect to processing and storing DOD sensitive information, preventing the device from touching a network with said information on it could be accomplished in future work on our application. Furthermore, locking out access to data storage via a future implementation of our application could meet the second of these requirements.

(26) STIG ID: SRG-MPOL-044

Rule:

> The organization must require that mobile devices used in facilities containing information systems processing, storing, or transmitting classified information, and the information stored on those devices, are subject to random reviews/inspections by organization defined security officials…A process of randomly inspecting or reviewing the various mobile devices, to include connected or imbedded capabilities, can be effective in ensuring compliance with the organization's mobile device policies and procedures.[131]

Fix:

> Develop and publish a requirement for mobile devices to be randomly reviewed/inspected for compliance with the organization's access control policy regarding the use of mobile devices within its facilities containing information systems processing, storing, or transmitting classified information, and the information stored on those devices.[132]

Discussion: The current version of our application provides a large font display of the current access state of WiFi, Bluetooth, Microphone, Mobile Data and Camera as either locked or unlocked. This feature was designed to meet the requirements of this STIG by enabling immediate feedback to any individual inspecting the device. The notion was that at any time a sailor could be required to unlock their device display and our application will be running with this display. The reader of the display has immediate feedback on the state of the features listed above. An example of one such inspection

[131] Defense Information Systems Agency, *Mobile Policy Security Requirements Guide* (SRG-MPOL-044), July 3, 2013, www.stigviewer.com/stig/mobile_policy_security_requirements_guide/2013-07-03/finding/V-35962, ID: V-35962.

[132] Ibid.

scenario discussed in later chapters would be watch turnover, whereby sailors are required to display their screen and log the status as either locked down or unlocked.

(27) STIG ID: SRG-MPOL-047

Rule:

> The organization must store and maintain a configuration baseline of each CMD, including application software…An integrity baseline scan must be maintained, so the baseline can be compared to any subsequent scan to identify any anomalies or determine if there are any security vulnerability trends or compromises to the system.[133]

Fix: "Maintain an integrity system baseline of the mobile device."[134]

Discussion: This is a perfect example of growth work for our application. Creating a baseline image of the device and delivering it to the command database could be part of application activation upon install and subsequently at each access to the quarterdeck or upon entering a specific space. Significant database architecture would have to exist based on the sheer number of devices on a ship or at a command. Consider a command with 300 personnel, and assume that each individual only has one device (which is an intentional gross underestimate). If they were to each provide a partial system image of 4GB over the command wireless network once per day, the database would have to be capable of handling and analyzing several hundred terabytes of data. The use of smaller partial scans and state of the art processing power might assist future baseline analysis.

(28) STIG ID: SRG-MPOL-053

Rule:

> The organization must establish a standard operating procedure (SOP) for data spills on CMDs…When a data spill occurs on a CMD, classified or sensitive data must be protected to prevent disclosure. After a data spill, the CMD must either be wiped using approved procedures, or destroyed if

[133] Defense Information Systems Agency, *Mobile Policy Security Requirements Guide* (SRG-MPOL-047), July 3, 2013, www.stigviewer.com/stig/mobile_policy_security_requirements_guide/2013-07-03/finding/V-35965, ID: V-35965.

[134] Ibid.

> no procedures are available, so classified or sensitive data is not exposed. If a data spill procedure is not published, the site may not use approved procedures to remediate after a data spill occurs and classified data could be exposed. This requirement also applies to sensitive DOD information stored on mobile OS devices that are not authorized to connect to DOD networks or store/process sensitive DOD information. Sensitive DOD data or information is defined as any data/information that has not been approved for public release by the site/Command Public Affairs Officer (PAO). In accordance with DOD policy, all components must establish Incident Handling and Response procedures. A classified message incident (CMI) or "data spill" occurs when a classified email or document is inadvertently sent on an unclassified network and received on a wireless email device. Classified information may also be transmitted through some other form of file transfer, to include web browser downloads and files transferred through tethered connections. CMDs are not authorized for processing classified data. The site's Incident Handling and Response procedures should reference National Security Agency/Central Security Service (NSA/CSS) Storage Device Declassification Manual 9–12, Section 5, for CMD destruction procedures.[135]
>
> Fix: "Create and publish an SOP for CMI on CMDs."[136]

Discussion: Our application currently utilizes device policy administration to restrict access to the camera. Another feature available to the device policy administrator (and in fact the user) is to remote wipe the device. We did not implement this feature out of fear of inadvertently wiping Space and Naval Warfare Systems Command (SPAWAR) research equipment, but the feature could be implemented in future iterations of our application. This tool could be utilized to remotely wipe a device in the event that a data spill is suspected, or evidence support the fact that it did occur, or if the device is lost/ misplaced. Also note that CMDs are now authorized for processing Secret data, with Top Secret currently under review.[137]

[135] Defense Information Systems Agency, *Mobile Policy Security Requirements Guide* (SRG-MPOL-053), July 3, 2013, www.stigviewer.com/stig/mobile_policy_security_requirements_guide/2013-07-03/finding/V-35971, ID: V-35971.

[136] Ibid.

[137] Defense Systems Information Agency, "DOD Mobility Classified Capability—Secret," accessed January 2, 2016, http://www.disa.mil/Enterprise-Services/Mobility/DMCC.

(29) STIG ID: SRG-MPOL-055

Rule:

> The organization must have a CMD Personal Use Policy that specifies what types of personal files are permitted on the device…Malware can be introduced to a DOD enclave via personally-owned applications and personal website accounts. In addition, sensitive DOD data could be exposed, altered, or exfiltrated by the same malware…The policy must include: (1) Installation of user-owned and free commercial applications, download of user-owned data (music files, picture files, etc.), (2) Connections to user social media accounts, (3) Use of geo-location aware applications that save or transmit the location of the device. The use of geo-location aware applications should be based on an Operational Security (OPSEC) risk assessment, (4) Connecting DOD managed mobile devices to personally-owned computers. (For example, a personally owned computer used to download personally-owned files to the mobile device).[138]

Fix: "Develop a Personal Use Policy which details the requirements for downloading user owned data (music files, picture files, etc.) on the mobile device."[139]

Discussion: Our application could be expanded to deny device access to other mobile applications such as Internet browsers, social media sites or apps or geolocation apps. Furthermore, our application could be configured to lock out and report to the command IS monitor any attempted or actual accesses to DOD computers or unauthorized networks or websites.

(30) STIG ID: SRG-MPOL-056

Rule: "The organization must have a CMD Personal Use Policy that specifies restrictions on the use of personal email…The DOD component must publish a Personal Use Policy for DOD component managed or owned CMDs."[140]

[138] Defense Information Systems Agency, *Mobile Policy Security Requirements Guide* (SRG-MPOL-055), July 3, 2013, www.stigviewer.com/stig/mobile_policy_security_requirements_guide/2013-07-03/finding/V-35973m V-35973.

[139] Ibid.

[140] Defense Information Systems Agency, *Mobile Policy Security Requirements Guide* (SRG-MPOL-056), July 3, 2013, www.stigviewer.com/stig/mobile_policy_security_requirements_guide/2013-07-03/finding/V-35974, ID: V-35974.

Fix: “Develop a Mobile Device Personal Use Policy which details the requirements for the operating system device to view or download personal email.”[141]

Discussion: As previously stated our application could be configured to prevent access to specific websites and or applications, thereby preventing access to restricted email accounts.

(31) STIG ID: SRG-MPOL-061

Rule: “The organization must establish standard operating procedures for provisioning mobile devices.”[142]

Fix: “Establish standard operating procedures for provisioning mobile devices to include integrity mechanisms protecting the confidentiality of over the air (OTA) provisioning.”[143]

Discussion: By controlling the security posture of a personal mobile device, there is no need for command provisioning of a mobile device.

(32) STIG ID: SRG-MPOL-063

Rule:

> Develop policy that states CMD software updates must only originate from DOD approved sources…Users must not accept over-the-air (OTA) wireless software updates from the wireless carrier or other non-DOD sources unless the updates have been tested and DOD approved. Unauthorized/unapproved software updates could include malware or cause a degradation of the security posture of the CMD and DOD network infrastructure.[144]

[141] Ibid.

[142] Defense Information Systems Agency, *Mobile Policy Security Requirements Guide* (SRG-MPOL-061), July 3, 2013, www.stigviewer.com/stig/mobile_policy_security_requirements_guide/2013-07-03/finding/V-35979, ID: V-35979.

[143] Defense Information Systems Agency, *Mobile Policy Security Requirements Guide* (SRG-MPOL-061), July 3, 2013, www.stigviewer.com/stig/mobile_policy_security_requirements_guide/2013-07-03/finding/V-35979, ID: V-35979.

[144] Defense Information Systems Agency, *Smartphone Policy Security Technical Implementation Guide* (WIR-SPP-008-02), October 9, 2012, www.stigviewer.com/stig/smartphone_policy/2012-10-09/finding/V-24964, ID: V-24964.

Fix: “Develop policy requiring CMD software updates originate from DOD approved sources.”[145]

Discussion: This STIG is directed at enterprise procured devices and does not directly relate to our research. Individuals would most likely prefer to have their software updates come from outside the DOD. Requiring that non-enterprise CMD’s receive their updates from DOD sources such as DISA could be part of the authorization to utilize the device on DOD Internet gateway network connections.

(33) STIG ID: SRG-MPOL-064

Rule: “The organizations DAA must approve the use of software PKI certificates on enterprise-activated CMDs prior to provisioning CMDs with DOD PKI digital certificates.”[146]

Fix: “Obtain DAA approval for the use of software certificates or purchase approved CAC readers for enterprise-activated CMDs.”[147]

Discussion: CAC readers for mobile devices are available for Android OS based mobile devices and iOS devices.[148] This presents a specific challenges for individuals with iOS devices, namely that they will have to have software-installed PKI certificates, which are only authorized with enterprise provided CMDs. Utilization of sites that require PKI authentication will therefore be limited to enterprise activated CMD’s based on STIG SRG-MPOL-064.

[145] Ibid.

[146] Defense Information Systems Agency, *Mobile Policy Security Requirements Guide* (SRG-MPOL-051), October 10, 2012, www.stigviewer.com/stig/mobile_policy_security_requirements_guide/2012-10-10/finding/SRG-MPOL-051, ID: SRG-MPOL-051.

[147] Ibid.

[148] Department of Defense, “CAC-Enabled Web Browsing Using the Thursby PKard Reader Smart Card Reader (SCR) and the Thursby PKard Reader Application [Quick Reference Guide].” Information Assurance Support Environment, accessed February 2, 0216, http://iase.disa.mil/pki-pke.

(34) STIG ID: SRG-MPOL-065

Rule:

> The organization must develop policy to restrict CMD Instant Messaging (IM) client applications to connect to only security-compliant, DOD-controlled IM servers…Non-DOD IM servers can be located anywhere in the world and may be under an adversary’s control. If a DOD CMD IM client connects to a non-DOD IM server, malware could be installed on the CMD from the server, or sensitive DOD data on the CMD could be transferred to the server.[149]

Fix: “Develop policy to require Instant Messaging (IM) client applications connect only to a security-compliant, DOD-controlled IM server.”[150]

Discussion: This is not applicable to personally owned devices, as the use of mobile IM applications is extensive. From Facebook, to snapchat, to google+, it would be unreasonable to require individuals to only connect to DOD IM servers. It is beyond the scope of our research to provide an IM solution for personal mobile devices.

(35) STIG ID: SRG-MPOL-066

Rule:

> The organization must obtain approval from the DAA or Command IT Configuration Control Board prior to installing a software application on a mobile device…Core applications are applications included in the CMD operating system. Applications added by the wireless carrier are not considered core applications. A security risk analysis must be performed by the DAA or DAA approval must be obtained prior to a mobile OS application being used. Non-approved applications can contain malware.[151]

[149] Defense Information Systems Agency, *Mobile Policy Security Requirements Guide* (SRG-MPOL-065), July 3, 2013, www.stigviewer.com/stig/mobile_policy_security_requirements_guide/2013-07-03/finding/V-35983, ID: V-35983.

[150] Ibid.

[151] Defense Information Systems Agency, *Mobile Policy Security Requirements Guide* (SRG-MPOL-066), July 3, 2013, https://www.stigviewer.com/stig/mobile_policy_security_requirements_guide/2013-07-03/finding/V-35984, ID: V-35984.

Fix: “Obtain DAA or Command IT CCB approval prior to installing non-core applications on CMDs.”[152]

Discussion: The application we developed as part of our research will be ultimately be submitted to DISA’s DOD Mobility Unclassified Capability (DMUC) Mobile Application Store (MAS) for evaluation and testing. DISA will serve as Designated Approving Authority (DAA) for our application.

(36) STIG ID: SRG-MPOL-067

Rule:

> The organization must perform a security risk analysis on a mobile operating system (OS) application by the DAA or DAA-authorized approval authority prior to the application being approved for use…Non-approved applications can contain malware. Approved applications should be reviewed and tested by the approving authority to ensure they do not contain malware, spyware, or have unexpected features (e.g., send private information to a website, track user actions, connect to a non-DOD management server). Core applications are applications included in the CMD operating system. Applications added by the wireless carrier are not considered core applications.[153]

Fix: “Perform a security risk analysis on a mobile operating system (OS) application prior to the application being approved for use.”[154]

Discussion: We have designed and built our application from the ground up in the Android Studio programming environment. No malware, spyware, or unexpected features have been coded into the application, and will be demonstrated in Chapters IV and V. The Navy will be required to perform a security risk analysis on the mobile device application we developed. With DISA approval through DMUC and MAS, and as the application was developed at Naval Postgraduate School the likelihood of the application containing malware, spyware or unexpected features is unlikely.

[152] Ibid.

[153] Defense Information Systems Agency, *Mobile Policy Security Requirements Guide* (SRG-MPOL-054), October 10, 2012, www.stigviewer.com/stig/mobile_policy_security_requirements_guide/2012-10-10/finding/SRG-MPOL-054, ID: SRG-MPOL-054.

[154] Ibid.

(37) STIG ID: SRG-MPOL-070

Rule:

> An authorization process must be developed and published that states the process to obtain approval before CMDs can connect to the organizations information system(s)…In order to protect their information systems, organizations must have a process in place ensuring mobile devices adhere to implementation guidance, meet published usage restrictions, and are processed through an authorization process prior to connecting to the information system(s). Lacking such a process, organizations will experience an array of unauthorized mobile devices, with a myriad of configuration settings and no usage restrictions, connecting to their information systems.[155]

Fix: "Develop and publish an authorization process to be performed on each mobile device before the device can connect to the organization's information system(s)."[156]

Discussion: This requirement could be met through the command check in and indoctrination process. The configuration settings required by different spaces within the command could be preset by the command security manager in a future implementation of our application. Note that we recommend specific settings be applied by the sensitivity of the space the individual will have access to, and will require routine updates to NFC tags to ensure maximum compliance and configuration control.

(38) STIG ID: SRG-MPOL-072

Rule:

> The organization must define locations the organization deems to be of significant risk to DOD information systems, in accordance with organizational policies and procedures…Failure of an organization to identify these locations could result in dangerous situations for its personnel, such as; damaged, stolen or compromised equipment; or

[155] Defense Information Systems Agency, *Mobile Policy Security Requirements Guide* (SRG-MPOL-070), July 3, 2013, www.stigviewer.com/stig/mobile_policy_security_requirements_guide/2013-07-03/finding/V-35988, ID: V-35988.

[156] Ibid.

> unauthorized access to, modification of, or destruction of sensitive or classified data.[157]

Fix: “Develop and document a list of high risk locations, and publish this list to security staff and other organizational personnel.”[158]

Discussion: This requirement should be met as part of annual information security training and with routine updates in the Plan of the Day (POD) so that all DOD members are aware of dangerous locations and organizations. A simple pop-up banner could be programmatically inserted into our application that informs users at a regular but non-intrusive interval about dangerous locations and organizations.

(39) STIG ID: SRG-MPOL-074

Rule: “The organization must apply organization defined inspection and preventative measures to mobile devices returning from locations the organization deems to be of significant risk to DOD information systems.”[159]

Fix: “Document the inspection and preventive measures applied to each mobile device returning from a high risk location, ensuring organization defined inspection and preventative measures are being applied.”[160]

Discussion: While we explored many programmatic solutions to this problem to automate reporting of cell phone location, such as GPS location log storage and delivery to command security management, this regulation is best handled via disclosure statements normally associated with high risk travel requests (Leave). Upon returning from travel, a mandatory stop on check-in could be the Information Assurance Manager for scan of the owners mobile device.

[157] Defense Information Systems Agency, *Mobile Policy Security Requirements Guide* (SRG-MPOL-072), January 24, 2013, www.stigviewer.com/stig/mobile_policy_security_requirements_guide/2013-01-24/finding/V-35990, ID: V-35990.

[158] Ibid.

[159] Defense Information Systems Agency, *Mobile Policy Security Requirements Guide* (SRG-MPOL-074), July 3, 2013, www.stigviewer.com/stig/mobile_policy_security_requirements_guide/2013-07-03/finding/V-35992, ID: V-35992.

[160] Ibid.

(40) STIG ID: SRG-MPOL-075

Rule:

> The organization must produce a written policy and training material that states CMDs that are classified as non-enterprise activated must not be used to send, receive, store, or process sensitive/FOUO or classified data and information or connect to DOD networks…Some CMDs are not authorized to store or process sensitive DOD data and information because they do not have required security controls to protect the data/information. There is a high risk that sensitive data will be exposed to unauthorized personnel with access to the device. Sensitive DOD data or information is defined as any data/information that has not been approved for public release by the site/Command Public Affairs Officer (PAO).[161]

Fix: "Develop a written policy and training material that states CMDs classified as non-enterprise activated must not be used to send, receive, store, or process sensitive/ FOUO or classified data and information or connect to DOD networks."[162]

Discussion: Similar to OPSEC training requirements examined later in this chapter, a pop-up or push notification banner requiring user input (pin) to pass could be developed such that on application initiation via NFC swipe the user certifies that they understand that their device will not access sensitive/FOUO information.

(41) STIG ID: SRG-MPOL-076

Rule:

> The organization must produce a written policy and training material that states CMDs classified as non-enterprise activated must not access DOD email systems…There is a high risk of introducing malware on a DOD email system or of compromising sensitive DOD data if these types of devices are connected to a DOD email system. There is a high risk sensitive data will be exposed to unauthorized personnel with access to the device if DOD email was viewed, processed, or stored on the device.[163]

[161] Defense Information Systems Agency, *Mobile Policy Security Requirements Guide* (SRG-MPOL-075), July 3, 2013, www.stigviewer.com/stig/mobile_policy_security_requirements_guide/2013-07-03/finding/V-35993, ID: V-35993.

[162] Ibid.

[163] Defense Information Systems Agency, *Mobile Policy Security Requirements Guide* (SRG-MPOL-076), January 24, 2013, www.stigviewer.com/stig/mobile_policy_security_requirements_guide/2013-01-24/finding/V-35994, ID: V-35994.

Fix: “Develop a written policy and training material that states CMDs classified as non-enterprise activated must not access DOD email systems.”[164]

Discussion: As stated above our application could be modified to prohibit access to DOD email either through webpage blocks or application blocks.

(42) STIG ID: SRG-MPOL-079

Rule:

> The organization must ensure all non-enterprise activated CMD users complete Operational Security (OPSEC) training that provides use guidelines and vulnerability mitigation techniques…Improper use of CMD devices can compromise both the CMD and the network, as well as, expose DOD data to unauthorized individuals. Without adequate OPSEC training, users are more likely to engage in behaviors that make DOD networks and information more vulnerable to security exploits. The security personnel and the site CMD device administrators must ensure non-enterprise activated CMD users receive OPSEC training.[165]

Fix: “Develop and publish policy mandating all non-enterprise activated CMD users complete Operational Security (OPSEC) training that provides use guidelines and vulnerability mitigation techniques.”[166]

Discussion: The required training could be conducted prior to application installation at command check-in, or during indoctrination briefs. This requirement could also be part of a push notification system built into future iterations of the application where by the device is placed in a fully locked down state until a security manager or information security officer certifies the training is complete and enters a unique pin into the device to certify that training is complete.

[164] Ibid.

[165] Defense Information Systems Agency, *Mobile Policy Security Requirements Guide* (SRG-MPOL-079), July 3, 2013, www.stigviewer.com/stig/mobile_policy_security_requirements_guide/2013-07-03/finding/V-35997, ID: V-35997.

[166] Ibid.

(43) STIG ID: SRG-MPOL-084

Rule: “The organization must secure all wireless network devices, such as wireless Intrusion Detection System (IDS) and wireless routers, access points, gateways, and controllers to prevent tampering or theft, or must be located in a secure room with limited access.”[167]

Fix: “Place all network devices (i.e., Intrusion Detection System (IDS), routers, Remote Access System (RAS), firewalls, etc.) in a secure room with limited access or otherwise secure to prevent tampering or theft.”[168]

Discussion: This STIG is not directly applicable to our research or application.

(44) STIG ID: SRG-MPOL-085

Rule: “The organization must ensure physical security controls are implemented for Secure WLAN (SWLAN) access points.”[169]

Fix: “Implement required physical security controls for the SWLAN.”[170]

Discussion: This STIG is not directly applicable to our research or application.

2. Commercial Mobile Device Policy Security Technical Implementation Guide: Release: 3 Benchmark Date: 12 Mar 2013

Rules one through six are CAT I severity related rules. The remainder are CAT II or III rules.

[167] Defense Information Systems Agency, *Mobile Policy Security Requirements Guide* (SRG-MPOL-084), July 3, 2013, www.stigviewer.com/stig/mobile_policy_security_requirements_guide/2013-07-03/finding/V-36002, ID: V-36002.

[168] Ibid.

[169] Defense Information Systems Agency, *Mobile Policy Security Requirements Guide* (SRG-MPOL-085), July 3, 2013, www.stigviewer.com/stig/mobile_policy_security_requirements_guide/2013-07-03/finding/V-36003, ID: V-36003.

[170] Ibid.

(1) STIG ID: WIR-SPP-003-01

Rule: “A data spill (Classified Message Incident (CMI)) procedure or policy must be published for site CMDs.”[171]

Fix: “Publish a Classified Message Incident (CMI) procedure or policy for the site.”[172]

Discussion: As discussed in STIG ID: SRG-MPOL-053, our application seeks to mitigate the occurrence of data spills by restricting access to data and data transfer when in lockdown.

(2) STIG ID: WIR-SPP-003-02

Rule:

> If a data spill (Classified Message Incident (CMI)) occurs on a wireless email device or system at a site, the site must follow required data spill procedures…This requirement also applies to sensitive DOD information stored on mobile OS devices that are not authorized to connect to DOD networks or store/process sensitive DOD information. Sensitive DOD data or information is defined as any data/information that has not been approved for public release by the site/Command Public Affairs Officer (PAO).[173]

Fix: “Follow required procedures after a data spill occurs.”[174]

Discussion: As discussed in STIG’s SRG-MPOL-053 and WIR-SPP-003-01 we recommend adding personal mobile device wiping authorization in exchange for DOD network access. This will allow wiping and thereby securing of personal mobile device’s suspected or found to have been party to a data spill.

[171] *Mobile Policy Security Requirements Guide* (WIR-SPP-003-01), July 3, 2013, www.stigviewer.com/stig/general_mobile_device_policy_non-enterprise_activated/2013-07-03/finding/V-24955, ID: V-24955.

[172] Ibid.

[173] Defense Information Systems Agency, *Smartphone Policy Security Technical Implementation Guide* (WIR-SPP-003-02), October 9, 2012, www.stigviewer.com/stig/smartphone_policy/2012-10-09/finding/V-24957, ID: V-24957.

[174] Ibid.

(3) STIG ID: WIR-SPP-005

Rule: “Mobile operating system (OS) based CMDs and systems must not be used to send, receive, store, or process classified messages unless specifically approved by NSA for such purposes and NSA approved transmission and storage methods are used.”[175]

Fix: “Publish written policy or training material stating CMDs must not process, send, or receive classified information unless approved for use.”[176]

Discussion: As stated in STIG ID: SRG-MPOL-075 a push notification with user feedback could notify and confirm user acceptance of policy with respect to classified data.

(4) STIG ID: WIR-SPP-009

Rule: “CMD Instant Messaging (IM) client application must connect only to a DOD controlled IM server compliant with the Instant Messaging STIG.”[177]

Fix: “Ensure the IM client application connects only to a DOD controlled IM server compliant with the Instant Messaging STIG.”[178]

Discussion: This STIG is nearly identical to STIG SRG-MPOL-065, and as stated therein providing IM server solutions is beyond the scope of our research.

[175] Defense Information Systems Agency, *Commercial Mobile Device (CMD) Policy Security Technical Implementation Guide (STIG)* (WIR-SPP-005), March 12, 2013, https://www.stigviewer.com/stig/commercial_mobile_device_cmd_policy/2013-03-12/finding/V-24960, ID: V-24960.

[176] Ibid.

[177] Defense Information Systems Agency, *Commercial Mobile Device (CMD) Policy Security Technical Implementation Guide (STIG)* (WIR-SPP-009), March 12, 2013, www.stigviewer.com/stig/commercial_mobile_device_cmd_policy/2013-03-12/finding/V-24965, ID: V-24965.

[178] Ibid.

(5) STIG ID: WIR-SPP-020

Rule:

> All non-core applications on the CMD must be approved by the DAA or the Command IT Configuration Control Board…Non-approved applications can contain malware. Approved applications should be reviewed and tested by the approving authority to ensure they do not contain malware, spyware, or have unexpected features (e.g., send private information to a website, track user actions, connect to a non-DOD management server).[179]

Fix: "Have DAA or Command IT CCB review and approve all non-core applications on mobile OS devices."[180]

Discussion: As this STIG is identical to STIG SRG-MPOL-066 our input remains the same. The application we developed as part of our research will be ultimately be submitted to DISA's DOD Mobility Unclassified Capability (DMUC) Mobile Application Store (MAS) for evaluation and testing. DISA will serve as designated approving authority (DAA) for our application.

(6) STIG ID: WIR-SPP-021

Rule:

> A security risk analysis must be performed on a mobile application by the DAA or DAA authorized authority prior to the application being approved for use…Core applications are applications included in the mobile device operating system. Applications added by the wireless carrier are not considered core applications. A security risk analysis must be performed by the DAA or DAA approved approval authority prior to a mobile OS application being approved for use.[181]

[179] Defense Information Systems Agency, *Commercial Mobile Device (CMD) Policy Security Technical Implementation Guide (STIG)* (WIR-SPP-020), March 12, 2013, www.stigviewer.com/stig/commercial_mobile_device_cmd_policy/2013-03-12/finding/V-32674, ID: V-32674.

[180] Ibid.

[181] Defense Information Systems Agency, *Commercial Mobile Device (CMD) Policy Security Technical Implementation Guide (STIG)* (WIR-SPP-021), March 12, 2013, www.stigviewer.com/stig/commercial_mobile_device_cmd_policy/2013-03-12/finding/V-32677, ID: V-32677.

Fix: “Have DAA or Command IT CCB use the required procedures to review mobile applications prior to approving them.”[182]

Discussion: Identical to STIG SRG-MPOL-067.

(7) STIG ID: WIR-SPP-001

Rule: “Site physical security policy must include a statement outlining whether CMDs with digital cameras (still and video) are permitted or prohibited on or in this DOD facility.”[183] Mobile devices with cameras are easily used to photograph sensitive information and areas if not addressed.

Fix: “Update the security documentation to include a statement outlining whether CMDs with digital cameras (still and video) are allowed in the facility.”[184]

Discussion: By design our application locks and unlocks camera (and therefore video) feature access. Since this is done by utilizing a device policy controller, the access is controlled at the root level. This design feature enables the local commander to set camera access permissions based on the site physical security policy discussed in the STIG. This design feature adds a second layer of defense against personnel using their camera’s in prohibited facilities or spaces with the facility.

(8) STIG ID: WIR-SPP-004

Rule:

> Required procedures must be followed for the disposal of CMDs…If appropriate procedures are not followed prior to disposal of a CMD, an adversary may be able to obtain sensitive DOD information or learn

[182] Ibid.

[183] Defense Information Systems Agency, *Commercial Mobile Device (CMD) Policy Security Technical Implementation Guide (STIG)* (WIR-WRA-003), March 12, 2013, www.stigviewer.com/stig/wireless_remote_access_policy_security_implementation_guide/2013-03-12/finding/V-25036, ID: V-25036.

[184] Ibid.

> aspects of the configuration of the device that might facilitate a subsequent attack.[185]

Fix: "Follow required procedures prior to disposing of a CMD or transitioning it to another user."[186]

Discussion: Based on the nature of our application and the notion that we have programmed it to be devoid of malware, maintaining version control over the source code is essential to avoid a third party adding unwanted features or malware. For this reason we recommend adding personal mobile devices to the disposal procedures related to CMD's. Personnel could take their device to the security manager or IAM and have the application removed or the device wiped depending on command preference.

(9) STIG ID: WIR-SPP-005

Rule: Mobile operating system (OS) based CMDs and systems must not be used to send, receive, store, or process classified messages unless specifically approved by NSA for such purposes and NSA approved transmission and storage methods are used.

Fix: Publish written policy or training material stating CMDs must not process, send, or receive classified information unless approved for use.

Discussion: As stated in STIG's WIR-SPP-005 and SRG-MPOL-075 push notifications could be sent to the user requiring feedback acknowledgement of policy with respect to transfer of classified data.

(10) STIG ID: WIR-SPP-010

Rule: "The site wireless policy or wireless remote access policy must include information on required CMD Wi-Fi security controls. The site wireless security policy or wireless remote access policy shall include information on locations where CMD Wi-

[185] Defense Information Systems Agency, *Commercial Mobile Device (CMD) Policy Security Technical Implementation Guide (STIG)* (WIR-SPP-004), March 12, 2013, www.stigviewer.com/stig/commercial_mobile_device_cmd_policy/2013-03-12/finding/V-24958, ID: V-24958.

[186] Ibid.

Fi access is approved or disapproved. The following locations will be specifically listed in the policy:

- Site-managed Wi-Fi access point connected to the NIPRNet (Enclave-NIPRNet Connected).
- Site-managed Wi-Fi access point connected to the Internet only (Internet Gateway Only Connection).
- Public Wi-Fi Hotspot.
- Hotel Wi-Fi Hotspot.
- Home Wi-Fi network (user managed).

Note: DOD CMD will not be used to connect to public or hotel Hotspots."[187]

Fix: "Publish CMD Wi-Fi security policy that includes information on required CMD Wi-Fi security controls."[188]

Discussion: This STIG provides excellent clarification on the types of Wi-Fi access points that are present within a typical DOD and Navy command, and how DOD procured CMD's may be used. Specifically, they note that DOD CMD's are not permitted on Public or Hotel Hotspots. Interestingly onboard ships with CANES systems, Wi-Fi hotspots are available in common spaces such as mess decks and wardroom. With minor modifications, our application could be used to lock out access to networks other than public or hotel hotspots, and completely lock out access to others (discussed further in Chapter VI).

(11) STIG ID: WIR-SPP-011

Rule:

> Mobile devices must be provisioned with DOD PKI digital certificates, so users can digitally sign and encrypt email notifications or other email

[187] Defense Information Systems Agency, *Commercial Mobile Device (CMD) Policy Security Technical Implementation Guide (STIG)* (WIR-SPP-010), March 12, 2013, www.stigviewer.com/stig/commercial_mobile_device_cmd_policy/2013-03-12/finding/V-24966, ID: V-24966.

[188] Ibid.

> messages required by DOD policy. DAA approval will be obtained prior to the use of software PKI certificates on mobile devices.[189]

Fix: "Obtain DAA approval for the use of software certificates or purchase approved CAC readers."[190]

Discussion: as discussed in STIG SRG-MPOL-064, provisioning of software PKI certificates is not authorized for personal mobile devices. Authorization is reserved for DOD enterprise provided CMD's.

(12) STIG ID: WIR-WRA-001

Rule:

> Users must receive training on required topics before they are authorized to access a DOD network via a wireless remote access device…Improper use of wireless remote access to a DOD network can compromise both the wireless client and the network, as well as, expose DOD data to unauthorized people.[191]

Fix: "Complete required training."[192]

Discussion: This is another example of an area where our application could amplify and add security layers to a STIG policy requirement. As stated many times throughout our analysis of the mobile application STIG's, push notification with user feedback acknowledging receipt of training could easily be added to our application.

[189] Defense Information Systems Agency, *Smartphone Policy Security Technical Implementation Guide* (WIR-SPP-011), October 9, 2012, www.stigviewer.com/stig/smartphone_policy/2012-10-09/finding/V-24968, ID: V-24968.

[190] Ibid.

[191] Defense Information Systems Agency, *Smartphone Policy Security Technical Implementation Guide* (WIR-WRA-001), February 2, 2012, www.stigviewer.com/stig/smartphone_policy/2012-02-02/finding/V-25034, ID: V-25034.

[192] Ibid.

(13) STIG ID: WIR-WRA-002

Rule: “The site must have a Wireless Remote Access Policy signed by the site DAA, commander, director, or other appropriate authority.”[193]

Fix: “Publish Wireless Remote Access Policy signed by the site DAA, commander, director, or other appropriate authority.”[194]

Discussion: This STIG is not within the scope of our research.

(14) STIG ID: WIR-WRA-003

Rule: “The site physical security policy must include a statement if CMDs with digital cameras (still and video) are permitted or prohibited on or in the DOD facility.”[195]

Fix: “Publish a site physical security policy that includes a statement if CMDs with cameras (still and video) are permitted or prohibited on or in the DOD facility.”[196]

Discussion: as discussed in STIG WIR-SPP-001, our application specifically locks out access to device camera functions. By doing so, our application adds a layer of security to this policy requirement, whereby access to a device camera can be accessed/ denied at device administration level.

[193] Defense Information Systems Agency, *Commercial Mobile Device (CMD) Policy Security Technical Implementation Guide (STIG)* (WIR-WRA-002), March 12, 2013, www.stigviewer.com/stig/wireless_remote_access_policy_security_implementation_guide/2013-03-12/finding/V-25035, ID: V-25035.

[194] Ibid.

[195] Defense Information Systems Agency, *Commercial Mobile Device (CMD) Policy Security Technical Implementation Guide (STIG)* (WIR-WRA-003), March 12, 2013, www.stigviewer.com/stig/wireless_remote_access_policy_security_implementation_guide/2013-03-12/finding/V-25036, ID: V-25036.

[196] Ibid.

(15) STIG ID: WIR-SPP-006-01

Rule: "Mobile device users must complete training on required content before being provided mobile devices or allowed access to DOD networks with a mobile device."[197]

Fix:

> This requirement applies to mobile operating system (OS) CMDs. All mobile device users must receive required training on the following topics before they are provided a mobile device or allowed access to DOD networks with a mobile device. Training is divided into two groups: Group A (general topics) and Group B (device specific topics).[198]

Discussion: The list of required training forms one of the most all-encompassing regulation pieces found in the STIG review. Coverage and integration of the guidance herein is highly encouraged for commanders, security officers and IA personnel as it provides an excellent framework for utilization of mobile devices. While the majority of the list covers enterprise issued CMDs, a number of the policy and procedure requirements could be tailored to include personally owned mobile devices. While outside the scope of our research, we identify the policy and procedure development and integration of personal mobile devices into these policy requirements. We present the topic lists in full to ensure full coverage of policy requirements for mobile devices:

Group A—General Topics

a. Requirement that personally-owned PEDs are not used to transmit, receive, store, or process DOD information unless approved by the DAA and the owner signs forfeiture agreement in case of a security incident.

b. Procedures for wireless device usage in and around classified processing areas.

c. Requirement that PEDs with digital cameras (still and video) are not allowed in any SCIF or other areas where classified documents or information is stored, transmitted, or processed.

d. Procedures for a data spill.

[197] Defense Information Systems Agency, *Mobile Policy Security Requirements Guide* (WIR-SPP-006-01), July 3, 2013, www.stigviewer.com/stig/general_mobile_device_policy_non-enterprise_activated/2013-07-03/finding/V-24961, ID: V-24961.

[198] Ibid.

e. Requirement that wireless email devices and systems are not used to send, receive, store, or process classified messages (does not apply to the SME PED).

f. Requirement that CMDs and systems will not be connected to classified DOD networks or information systems.

g. Requirement that a user immediately notify appropriate site contacts (i.e., IAO, CMD management server administrator, supervisor, etc.) when his/ her CMD has been lost or stolen.

h. Secure Bluetooth Smart Card Reader (SCR) usage:

- Secure pairing procedures.
- Perform secure pairing immediately after the SCR is reset.
- Accept only Bluetooth connection requests from devices they control.
- Monitor Bluetooth connection requests and activity in order to detect possible attacks and unauthorized activity.

i. Procedures on how to sign and encrypt email.

j. If Short Message Service (SMS) and/or Multi-media Messaging Service (MMS) are used, IA awareness training material should include SMS/ MMS security issues.

k. Requirement that Over-The-Air (OTA) wireless software updates should only come from DOD approved sources.

l. When CMD Wi-Fi Service is used, the following training will be completed:

- Procedures for setting up a secure Wi-Fi connection and verifying the active connection is to a known access point.
- Approved connection options (i.e., enterprise, home, etc.).
- Requirements for home Wi-Fi connections.
- The Wi-Fi radio will be disabled by the user whenever a Wi-Fi connection is not being used.
- The Wi-Fi radio must never be enabled while the CMD is connected to a PC.

m. Do not discuss sensitive or classified information on non-secure (devices not FIPS 140–2 certified or NSA Type-1 certified for voice) cellular phones, cordless phones, and two-way radios used for voice communications.

n. Do not connect PDAs, smartphones, and tablets to any workstation that stores, processes, or transmits classified data. (Exception: SME PED).

o. The installation of user owned applications, including geo-location aware applications, on the mobile device will be based on the Command's Mobile Device Personal Use Policy.

p. The use of the mobile OS device to view and/or download personal email will be based the Command's Mobile Device Personal Use Policy.

q. The download of user owned data (music files, picture files, etc.) on the mobile device will be based the Command's Mobile Device Personal Use Policy.

r. The use of the mobile device to connect to user social media web accounts will be based the Command's Mobile Device Personal Use Policy.

s. When the Bluetooth radio is authorized for use with an approved smartcard reader or handsfree headset, the user will disable the Bluetooth radio whenever a Bluetooth connection is not being used.

t. All radios on the mobile device (Wi-Fi, Bluetooth, near-field communications (NFC)) must be turned off when not needed.

u. Procedure on how to disable Location Services on the device. Location Services must be disabled for all applications or enabled only for applications approved by the DAA for location based services.

Group B—Device Specific Topics

Additional BlackBerry requirements:

a. If the use of the BlackBerry Keeper is approved by the DAA, users are trained on password configuration and change requirements. Passwords must be changed at least every 90 days

b. When SCR is used with a PC, users with PC administrative rights will not disable the RIM Bluetooth Lockdown tool on the PC.

c. When using an approved Bluetooth headset or handsfree device the following procedures will be followed:

- The user will pair only an approved device to the BlackBerry handheld.
- If the user receives a request for Bluetooth pairing on their BlackBerry handheld from a Bluetooth device other than their smart card reader (CAC reader) or headset, the request will not be accepted by the user.
- Pairing of a Bluetooth headset with the BlackBerry handheld will be completed in a non-public area whenever possible.

Additional iOS device (iPhone and iPad) requirements:

a. Procedure on how to disable the device Bluetooth radio when not being used.

b. Procedure on how to disable the device Wi-Fi radio when not being used.

c. Procedure to disable “Ask to Join Networks” Wi-Fi feature. This feature must be disabled at all times.

d. Message should be considered an unsecure messaging application, similar to cellular SMS. Sensitive information should not be sent via iMessage.

e. Procedure for not allowing applications access to PIM date (calendar, address book, etc.) when prompted during application install. The only allowed exception is for the secure email application (for example, the Good application).

f. Procedure for not allowing applications access to iOS device Personal Information Manager (PIM) data (calendar, contacts, notes, etc.) when prompted during application installation. The only allowed exception is for the DOD email application (for example, the Good Technology app).

Additional Android requirements:

a. Procedure on how to disable the device Bluetooth radio when not being used.

b. Procedure on how to disable the device Wi-Fi radio when not being used.

Additional training requirements for mobile device not authorized to connect to a DOD network or store/process sensitive DOD information (Non-Enterprise activated):

a. Mobile Device (Non-Enterprise Activated) must not be connected to a DOD wired or wireless network. Allowed exception: the device can be connected to a DOD managed Internet-Gateway-only connected Wi-Fi access point (AP).

b. Mobile Device (Non-Enterprise Activated) must not have sensitive or classified data stored or processed on the device.

c. Mobile Device (Non-Enterprise Activated) must not be used to connect to a DOD email system.

d. The user will read and be familiar with the local site and/or Command must publish a Personal Use Policy for site/Command managed or owned CMDs.

Additional BlackBerry Playbook Tablet requirements:

When using BlackBerry Bridge, the user will not attach files saved on the Playbook to email messages sent on the BlackBerry smartphone.

Note: Listing training requirements in the User Agreement is an acceptable procedure for informing/training users on many of the required training topics.

(16) STIG ID: WIR-SPP-006-02

Rule: “Mobile users must complete required training annually.”[199]

Fix: “Complete required training annually for all CMD users.”[200]

Discussion: STIG WIR-SPP-006-02 provides the full and most comprehensive list for annual training. As previously mentioned our application could be modified to provide banner notification of training requirements such as due date, date last completed, number of days to report completion etc. Furthermore, these banners could be designed to require user feedback to clear them whereby consent to the requirements, dates, etc., are acknowledged.

(17) STIG ID: WIR-SPP-007-01

Rule: “The site Incident Response Plan or other procedure must include procedures to follow when a mobile operating system (OS) based mobile device is reported lost or stolen.”[201]

Fix:

> Publish procedures to follow if a mobile operating system (OS) based CMD is lost or stolen. Mobile device user notifies IAO, SM, and other site personnel, as required by the site’s incident response plan, within the timeframe required by the site’s incident response plan. The IAO notifies the mobile device management server system administrator and other site personnel, as required by the site’s Incident Response Plan, within the timeframe required by the site’s Incident Response Plan. The site mobile device management server administrator sends a wipe command to the CMD and then disables the user account on the management server or

[199] Defense Information Systems Agency, *Smartphone Policy Security Technical Implementation Guide* (WIR-SPP-006-02), October 9, 2012, https://www.stigviewer.com/stig/smartphone_policy/2012-10-09/finding/V-28317, ID: V-28317.

[200] Ibid.

[201] Defense Information Systems Agency, *CMD Management Server Policy Security Technical Implementation Guide* (WIR-SPP-007-01), August 5, 2014, https://www.stigviewer.com/stig/cmd_management_server_policy/2014-08-05/finding/V-24962, ID: V-24962.

> removes the CMD from the user account. The site will contact the carrier to have the device deactivated on the carrier's network.[202]

Discussion: The importance of reporting lost or stolen devices cannot be understated. In my experience in the fleet, numerous personal mobile devices were lost or stolen in foreign ports or while at sea resulting theft of personal financial data and personally identifiable information resulting in thousands of dollars of loss and hundreds of hours of man hours to repair or recover. The overall loss of skilled operators and their resultant degradation of mission effectiveness was significant. For these reasons we recommend personal mobile devices be included in some tracking system and when lost, require reporting of the lost device. For example, what if unclassified personnel and ships' movements were stored on the lost device? If those devices were tracked, and were able to be remotely wiped, the data may not have the potential to fall into an adversary's hands.

(18) STIG ID: WIR-SPP-007-02

Rule: "Required actions must be followed at the site when a CMD has been lost or stolen."[203]

Fix: "Follow required actions when a CMD is reported lost or stolen."[204]

Discussion: As mentioned above, based on the general sensitivity of personal data stored on everyone's mobile devices, we recommend personal mobile devices be registered with the command security manager, IAO etc., so that a remote wipe can be initiated by our application in cases of loss or theft. While anticipated resistance to such a control mechanism is high, the resultant protection of personal data would be invaluable when measured against the amount of added sailor response time and resultant mission degradation when personal devices are lost or stolen.

[202] Ibid.

[203] Defense Information Systems Agency, *Wireless Management Server Policy Security Technical Implementation Guide* (WIR-SPP-007-02), September 30, 2011, https://www.stigviewer.com/stig/wireless_management_server_policy/2011-09-30/MAC-1_Classified/xml, ID: V-24969.

[204] Ibid.

(19) STIG ID: WIR-SPP-008-01

Rule:

> The mobile device SA must perform a wipe command on all new or reissued CMDs and a STIG-compliant IT policy will be pushed to the device before issuing it to DOD personnel. The CMD system administrator must perform a wipe command on all new or reissued CMDs, reload system software, and load a STIG-compliant security policy on the CMD before issuing it to DOD personnel and placing the device on a DOD network. The intent is to return the device to the factory state before the DOD software baseline is installed.[205]

Fix: "Perform a wipe command on all new or reissued mobile devices."[206]

Discussion: Not directly related to our research or application.

(20) STIG ID: WIR-SPP-008-02

Rule:

> Mobile device software updates must only originate from approved DOD sources. CMD system administrators should push OTA software updates from the CMD management server, when this feature is available. Otherwise the site administrator should verify the non-DOD source of the update has been approved by IT management.[207]

Fix: "Ensure CMD software updates originate from DOD sources or approved non-DOD sources only. Users do not accept Over-The-Air (OTA) wireless software updates from non-approved sources."[208]

Discussion: Already addressed in SRG-MPOL-063 this STIG is directed at enterprise procured devices and does not directly relate to our research. Individuals would most likely prefer to have their software updates come from outside the DOD.

[205] Defense Information Systems Agency, *Smartphone Policy Security Technical Implementation Guide* (WIR-SPP-008-01), November 28, 2011, https://www.stigviewer.com/stig/smartphone_policy/2011-11-28/MAC-3_Sensitive/xml, ID: V-24963.

[206] Ibid.

[207] Defense Information Systems Agency, *Smartphone Policy Security Technical Implementation Guide* (WIR-SPP-008-02), November 28, 2011, https://www.stigviewer.com/stig/smartphone_policy/2011-11-28/MAC-3_Sensitive/xml, ID: V-24964.

[208] Ibid.

Requiring that non-enterprise CMD’s receive their updates from DOD sources such as DISA could be part of the authorization to utilize the device on DOD Internet gateway network connections.

III. TECHNOLOGICAL ASPECTS AND CONSIDERATIONS

The technological aspects of recommending a lockdown or security program for any operating system must consider the methods of implementing control mechanisms. Some forms of input could give greater control to an account administrator while at the same time opening the operating system up to greater forms of attack or granting privileges with unintended accesses. It is important that any implementation follows the recommendations of the OS developer, specifically Google in our case, and not try to use unintended methods that will be removed in future updates. In this chapter, we examine two input methods for starting an application or exchanging data and then examined some of the technologies that comes bundled on typical smart devices. Looking at each will give the reader of this thesis a small piece of the information that should be considered when discussing these technologies and how they work.

The first input method we consider is Quick Response (QR) codes. Through our research we determine that they are not the best fit for our application but they may have a place in future versions of the program. The second form of input we examine is near field communication (NFC), which allows for programmable, quickly customizable tags to perform actions on a smart device. QR codes offer direct access to data through a smart device's camera while NFC allows for RF transmission of programmed data. An introduction to how each works and the input method's characteristics are discussed along with potential vulnerabilities. Additional mitigations are be recommended where appropriate and implementations will be built around scenarios so that the benefits can be imagined.

Looking at the device as a whole makes it difficult to decide where to start locking down features. However, as shown in the documentation review, there are several specific pieces of hardware that would need to be disabled in an environment where official DOD work is being conducted. Looking at each piece individually, an application that can lock down a device can start to take shape. Each of these components is discussed along with some of the ways in which data is moved on and off the device. While our application deals mostly with the hardware lockdown implications, we would like a method to control the ability to send and receive cellular data in the current Android API. Future versions should consider methods to disable the sending of text messages over standard cellular networks and the potential for authorized Wi-Fi access to a ship’s intranet. For now, however, we demonstrate what should be and can be disabled prior to adding increased flexibility.

A. INPUT METHODS

When designing our application in the early stages we were unsure about a method to initialize the lockout features on the phone. Trusting a sailor to start an application and push a button did not provide a suitable solution and the intent of the application was to put the lockout control in the hands of the security manager. We ultimately look at Quick Response codes and near field communication as input methods to initialize our application. A discussion on the characteristics of each will include strengths and weaknesses.

1. QR Code Characteristics and Uses

Scanning of 1D barcodes for quick access to data started to be used in the 1960s with railroad companies.[209] They proved useful, but were not durable enough for long-term use and were eventually abandoned with the exception of a few manufacturing environments. It was not until the supermarket industry adopted them for point-of-sale purchases that a greater audience started taking note of their potential for carrying data and automating processes on a grander scale. Having proven their utility to the public, it was in this era that the use of scanning tech expanded and eventually a need for higher capacity was needed in the coding scheme.

2D and Quick Response (QR) codes have been used in the industry sector for several years. A QR code for any implementation is a product of a library, which is referenced when it is scanned, the amount of data written, and the amount of error correction embedded in the code. The reference library can be written or purchased online but is necessary to encode and decode the actual QR data. Sites exist to give

[209] Tony Seidman, “Barcode History: Barcodes Sweep the World,” Barcoding Incorporated, accessed January 20, 2016, http://www.barcoding.com/information/barcode_history.shtml.

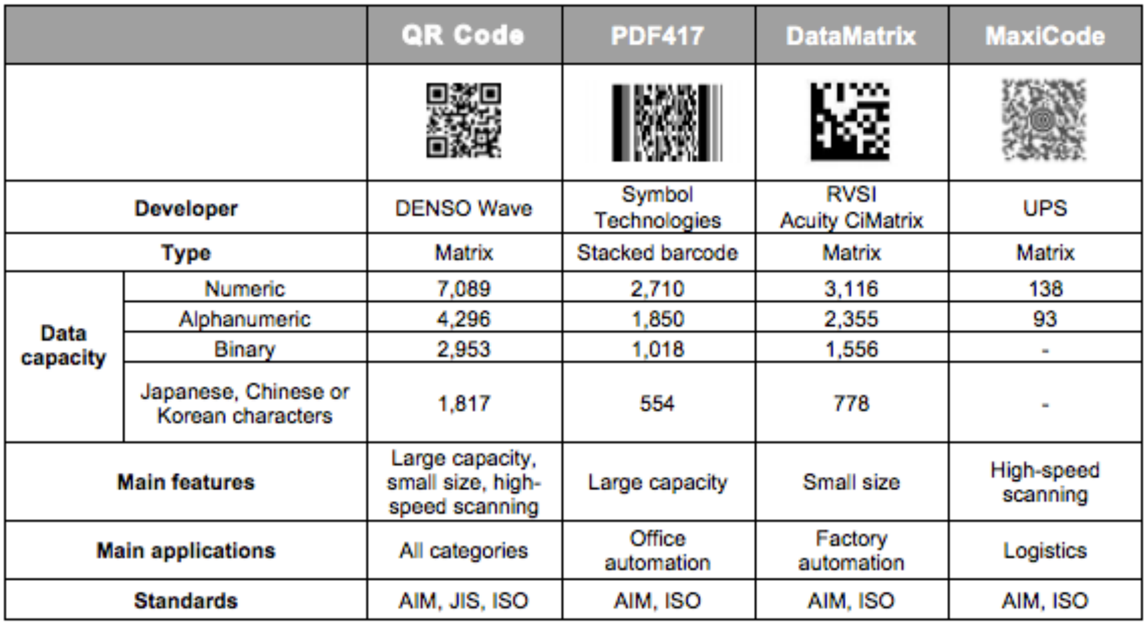
example QR code libraries if desired, but the development of a library falls outside the scope of this thesis and would require a greater amount of upfront effort to create and secure.

The amount of data found in a code is a very important component to understand as it can range from a few alpha-numeric characters to 4300 characters (Figure 9).[210] This is a significant amount of data and can point to sites on a network or can trigger small code implementations already present on a device. An example of this would be in the restaurant industry where companies can use a QR code to jump to a company's menu hosted online or their application in any operating system's app store. At that location, executable code can exist that will allow a customer to order and pay for food while waiting or before arrival. This can be translated to virtually any requisition system where a worker would need to access a database and submit an order for replacement parts or consumables.

By design, QR codes are resistant to the impacts of damage and debris. Their utility in dirty, manufacturing-type environments has been proven and provided workers a method to quickly implement various controls. These include ordering parts, documenting movement along patrol routes, or the ability to access needed documents and maintenance/procurement requirements. QR codes have been used in office environments to automate processes like network management, inventory controls, and restocking mechanisms. Marketing companies now use them to point to websites of interest to consumers. Movie posters, safety notices, and areas where additional info can be presented on a mobile device can and do utilize QR codes to provide access to collateral information through websites and online forums. QR codes are easily implemented in many environments, relatively customizable, provide quick and easy access to data, and their functionality is ultimately limited by the type of library referenced and the writable data limits within the code itself.

[210] DENSO ADC, *QR Code Essential* (White Paper R1f), 2011, http://www.nacs.org/LinkClick.aspx?fileticket=D1FpVAvvJuo=&tabid=1426&mid=4802.

Figure 9. A comparison of QR and 2D barcode data storage limits

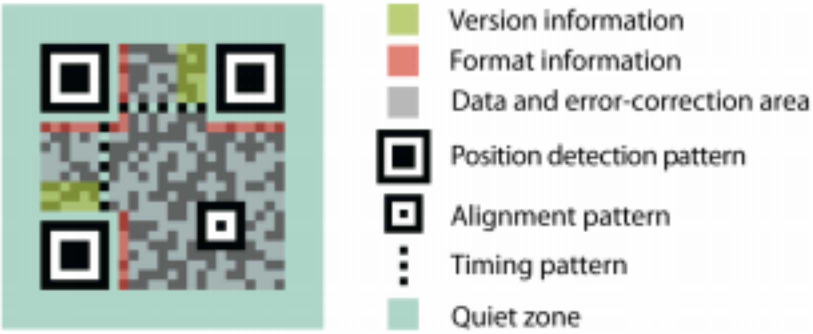


		QR Code	PDF417	DataMatrix	MaxiCode
Developer		DENSO Wave	Symbol Technologies	RVSI Acuity CiMatrix	UPS
Type		Matrix	Stacked barcode	Matrix	Matrix
Data capacity	Numeric	7,089	2,710	3,116	138
	Alphanumeric	4,296	1,850	2,355	93
	Binary	2,953	1,018	1,556	-
	Japanese, Chinese or Korean characters	1,817	554	778	-
Main features		Large capacity, small size, high-speed scanning	Large capacity	Small size	High-speed scanning
Main applications		All categories	Office automation	Factory automation	Logistics
Standards		AIM, JIS, ISO	AIM, ISO	AIM, ISO	AIM, ISO

Source: DENSO ADC, *QR Code Essential* (White Paper R1f), 2011,://www.nacs.org/ LinkClick.aspx?fileticket=D1FpVAvvJuo=&tabid=1426&mid=4802, 3.

QR codes were developed with ease of functionality in mind and have given users the ability to control the amount of error correction for lost/damaged data within the code, in addition to scanability in virtually any orientation. The error correction in a code can be increased up to 30 percent at the expense of writable data, however this is a very useful feature in environments where dirt and debris could cover a code or damage could affect the code's surface. In addition to allowing for various levels of code obfuscation, the error correction allows for code distortion. Placing a code does not necessarily require a flat surface as the data can be maintained with the formatting and timing patterns printed into the code. Finally, QR code orientation is also flexible since the codes themselves have built in position detection patterns. These allow a scanner to quickly determine where the data in a code will start and stop and eliminates the need of scanning from various sides and angles until the data makes sense and is useful (as was the case with traditional 2D codes). All of these features combined keep the need to reprint and replace codes to a minimum and still provides utility in normal operations. Controlling the data against the level of desired correction gives users a significant control over the actual code's size when printed and orientation patterns allow for more flexibility in surface placement (see Figure 10).[211]

Figure 10. A depiction of data storage format on a QR format



Source: DENSO ADC, *QR Code Essential*, 4.

2. QR Code Weaknesses and Vulnerabilities

Despite their versatility QR codes still suffer from many weaknesses. Even with error correction, data can be lost if the code is smudged, ripped, or covered beyond the correction threshold. This will make a code useless and require that it is physically replaced. Additionally, if the data being referenced changes then the code must be changed to direct data appropriately as there is no way to push updated data to an old QR code. If the codes of a facility point to specific intranet sites, and that network is reconfigured, it is possible that every QR code will have to be reprinted and replaced if the library cannot be made to compensate for the change.

With regard to replicability, it is entirely possible that a malicious actor could replace codes with bad information that redirects to sites with bad data or set up to exploit vulnerabilities. User interaction could be required after the redirect, but it is not out of the realm of imagination to suppose a user would accept whatever prompts come from a

[211] Ibid.

supposedly trusted code. This opens the door to exploit OS vulnerabilities and goes beyond the inconvenience of just having to replace all distributed codes.

With the advent of smart devices and their onboard cameras, QR codes have been brought to the general public. Their successful adoption is a topic of much debate among marketing companies but their flexibility cannot be denied.[212] Manufacturers, employers, and marketers can easily encode data for specific applications. Individuals scanning the codes have an innate trust that the data being scanned is as safe as the data on their device. It is this feeling that can give the QR code its greatest vulnerability for abuse.

As is the case with many technologies related to smart devices or secure environments, it is the end user who presents the greatest opportunity to exploit a vulnerability. The example mentioned before only touched on replacement of codes in a trusted environment. However, consider a user that installs an application with other malicious designs. This application could collect data about a user, where codes are scanned, network options at the time of scan, and many specifics about what is read. Once in an open network again, that data could be forwarded and provide usable information.

The ability to embed executable code in QR codes present another form of weakness. If that code allows a mobile device to send data back to a host, then it is a matter of what data is collected and sent by the code. An example of this was documented in a 2013 article, "What Is behind That QR Code?" The authors referenced various examples of when commands executed through a user reading a QR code allowed access to personal data, device data, or device commands such as a device reset.[213]

A QR code in a controlled environment with proper policy and implementation could allow for quick access to vast amounts of information. On a restricted network, they could enable a soldier or sailor access to repair publications, placing part orders, or

[212] Brian Morris, "Are QR Codes Thriving or Dying?" Business 2 Community, May 21, 2015, http://www.business2community.com/marketing/qr-codes-thriving-dying-01228016#DoWpfBHtY3rYivO3.97.

[213] InfoSec Institute, "What Is behind That QR Code?" March 21, 2013, http://resources.infosecinstitute.com/what-is-behind-that-qr-code/.

completing necessary training with a simple scan of a code and an on-device camera. Monitored implementation and policies that control their use can help mitigate the listed vulnerabilities but one must keep in mind there is no built-in security features with a QR code. The potential for good can outweigh the risks on an unclassified network where the stored data will not cause harm if it is recovered, but appropriate user training should be provided along with policy to reduce the risk of exposing personal smart devices to malware.

3. NFC Characteristics and Uses

Near field communication (NFC) is a technology based off radio frequency identification (RFID). RFID has also been in place for several years and has been used in contactless card readers for identification and area admission purposes throughout the DOD. Badges at secure facilities have relied on this technology to securely identify cardholders by having encrypted data on a card that is coupled with a pin chosen by the card holder. These two items together (something a user has and something a user knows) provides a form of two-factor authentication before someone can enter a facility.

The concept of powering a passive device, that is one without its own source of power, has been shown to be useful. As with other RFID technology, NFC will work between a user’s mobile device (typically a phone or tablet) and a passive tag that stores some limited amount of data. When the mobile device is placed in close proximity to the tag, a magnetic flux field is generated, the passive tag has power, and the data written can be read by the mobile device. Once read, the mobile device can use the data for setup of applications, event triggering, data storage, payments, or various other features. The flexibility provided by smart mobile technologies and NFC can allow for very specific access control mechanisms and controller events on a mobile device. It is with this in mind that focus shifted away from strictly using the QR code as an input source.

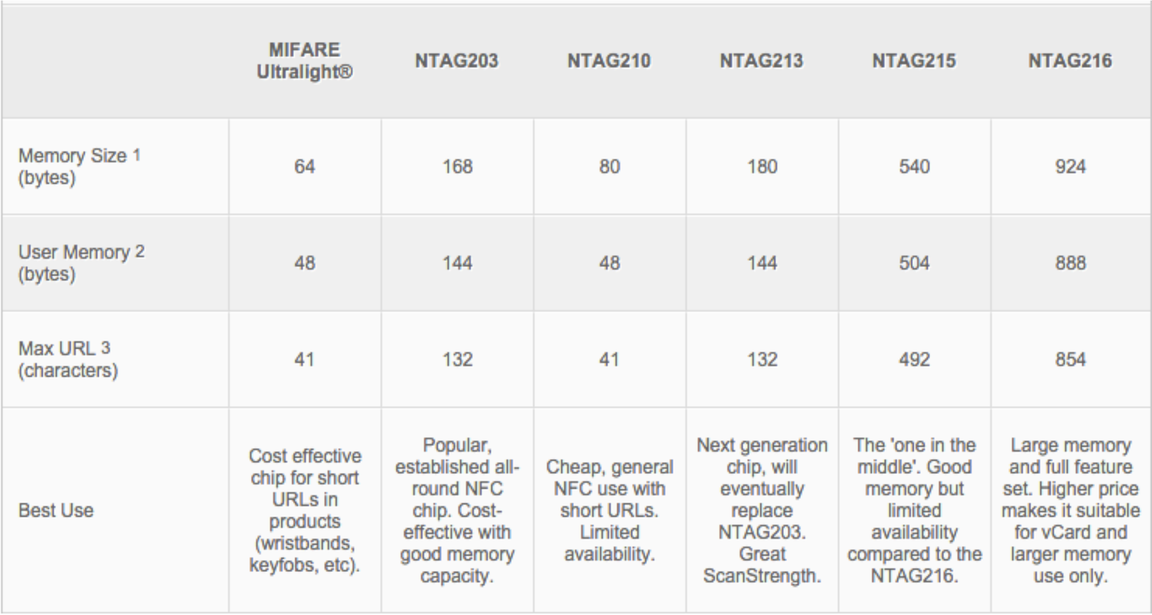
NFC devices are typically more robust than QR codes for a variety of reasons. The largest advantages are that there does not have to be a direct line of site between a tag and a reader and security options are in place. A tag or NFC device placed in a secure container can still exchange data if the minimum distance requirements are met. An NFC tag does not become smudged or torn in an industrial environment and require a reprint and replacement. Since it can be internally attached and still exchange data, it can operate in even harsher industrial environments than a QR code. This is not to say that NFC will always be preferred, rather it is to say that the environment and purpose of each technology should be considered before a design is implemented. If data is only available from a distance of several feet, a larger, printed QR code would be preferable to an NFC device that is not readable by a user.

Since NFC is a device with a chip and memory, there are some security and data encryption options that can be used with them. This level of security is not provided in a QR code. The security provided by having a writable chip does come at a cost that is invested up front to get the appropriate hardware for NFC and then the continued cost of buying NFC tags. While tags have come down in price over time, it will always have a higher cost associated than a QR code. QR code replacement only requires that a security manager set up the code and print it. Reprinting codes is quick and cheap, while NFC's security and reliability come at a higher cost.

The average NFC tag offers less user storage than can be coded into a QR code but there are options so that with proper planning the right tags can be implemented (see Figure 11). A peer-to-peer communication option also exists so that NFC devices, which are powered and have larger storage can communicate. This is done between mobile devices, payment systems, etc. Setting up an NFC to initiate an application on a mobile device requires little physical memory and in the case of this thesis the smallest memory options were used to provide examples. As security is implemented and larger settings are desired it will be necessary to allow for more programming space and more expensive tags to solve more complex problems.

Keeping these considerations in mind as we discuss usage scenarios in an official context will assist in imagining the full potential of BYOD environments for these two particular input methods. With appropriate construction and the proper security policies in place both can be useful in a command environment. The documentation and critiques of both technologies and their lack of utility is often discussed from a consumer perspective. However, practical implementations exist in the DOD and when coupled with a BYOD strategy of locking down a mobile device, these technologies can allow for future flexibility and utility.

Figure 11. Comparison of NFC tag storage values and uses



	MIFARE Ultralight®	NTAG203	NTAG210	NTAG213	NTAG215	NTAG216
Memory Size 1 (bytes)	64	168	80	180	540	924
User Memory 2 (bytes)	48	144	48	144	504	888
Max URL 3 (characters)	41	132	41	132	492	854
Best Use	Cost effective chip for short URLs in products (wristbands, keyfobs, etc).	Popular, established all-round NFC chip. Cost-effective with good memory capacity.	Cheap, general NFC use with short URLs. Limited availability.	Next generation chip, will eventually replace NTAG203. Great ScanStrength.	The 'one in the middle'. Good memory but limited availability compared to the NTAG216.	Large memory and full feature set. Higher price makes it suitable for vCard and larger memory use only.

This graphic adapted from NFC chip supplier RapidNFC. Source: "Which NFC Chip?" accessed January 10, 2016. http://rapidnfc.com/which_nfc_chip.

4. NFC Weaknesses and Vulnerabilities

NFC lives in the domain of radio frequency identification (RFID) standards and allows for short-range communications and pairing between devices. This exchange of information is almost instant once a device is in range of a reader tag. While the expectation is that a user would not scan a tag that was unknown, it is possible for misuse to expose a device to input vulnerabilities mentioned in the QR code section. A tag could prompt a device to open a webpage with malicious data, prompt the download of a third party application, or simply not perform the desired action and leave the user unaware that a setting had been changed.

Additionally, there is no detection method to see if the data exchange is being monitored. If an NFC tag were to communicate securely with a mobile device, it is not certain that radio exchange was not recorded for future use. A mobile device that is already compromised could be used to scan any tag nearby and record the actions or data exchanged. This could give an attacker insight into how an NFC tag is coded, information about provided encryption, and details about how to work within an established system. This particular issue is inherent in any RF system and could be especially present when users carry their personal devices outside of the controlled environment on a ship. It is the activity outside of DOD networks and what that means for mobile devices that creates the largest area of vulnerability and would have to be addressed in other device policies.

B. SOFTWARE APPLICATION LOCKDOWN

Using an application to lockdown features and control settings on a cellular device requires addressing each input method with coding specific to it. Disabling the camera is not done in the same manner as disabling Wi-Fi or cutting off the microphone. Several applications were looked at to determine which approaches were the most effective and would function properly on the most devices. Additionally, ensuring that our particular choices for implementation were usable on the most devices required careful consideration. Device control based on time, location detection, and root-level control features have worked its way into applications using Android's administrative control methods coupled with standard accesses. These give a programmer greater flexibility over what the application can do and access, and forces the user to grant permission for the application to function properly. While our application is a basic proof of concept, there is a lot of room for flexibility and increased protection/detection methods for Android devices.

Even with the more robust device administration control system that was introduced in Android 22, there are still capabilities and hardware components that are not easily disabled or mitigated through standard practices.[214] There may come a time when direct control of all input/output methods is streamlined in device administration settings. However, Android currently has not provided a default control for writing data to various forms of memory or directly disabling text input, and these are just two

[214] Developers, "Device Administration," accessed February 14, 2016, http://developer.android.com/guide/topics/admin/device-admin.html.

examples. Explanations will be given below, but some of these will have to be creatively mitigated through more robust coding in the future or they may require designs that were not intended by Google (which could be patched out of a future build of Android). Our application looks to work within the intended uses of Android and the recommended methods for coding applications to reduce the chance that our implementation could simply be patched away in a future update.

The final consideration when developing this application is that our prototype does not have a file in which device state is saved. This is possible to do, but requires more robust programming than the scope of this thesis. State is saved locally on the Android device, but additional features will require a file within the app that saves and is accessed for crash/restart re-enabling of the lockdown. Additionally, the application does not automatically restart when it crashes or is disabled. It is recommended that if a form of this is implemented in the future, the application is designed to restart after it is cut off while locking down features. For this to work, it should reference a state file maintained inside the application and use that to ensure that the device is placed in a state similar to when the application stopped running. This would be especially important for features not controlled by device administrator policy settings (for example, the microphone, Wi-Fi, or Bluetooth states) that could be potentially turned back on by a user once the application is turned off.

This thesis touches on the concern associated with third party application access and what that could mean for using a device in a secure environment. However, the insider threat (intentional or accidental) continues to be the initial concern with non-enterprise issued mobile devices. A device that can be managed with regard to the various input/output methods in a lightweight application is a reasonable goal. Building on this and adding plug ins or other approved applications to increase functionality, encrypted data separation, and utility to the user and DOD would pave the path for a useful BYOD implementation. When coupled with appropriate policy, user training, and eventual network controls, a device lockdown application could provide an onboard software solution to device hardware concerns.

1. Camera Considerations

With regard to onboard hardware, cameras offer a simple method to store large, detailed amounts of data in memory in a quick, discrete manner. While carrying a mobile device inside an unclassified facility, data can be visually documented regarding layout, personnel, guards, and other useful information. Before allowing a mobile device into a facility that could have official use areas, or specifically the spaces on a ship, all cameras should be collected or turned off and verified.

The question then turns to how a phone with a camera on board can be made to disable the camera's functionality completely. On studying this, we found that early implementations of camera disabling applications simply took control of the camera and never released it. This actually exploits the design of the Android OS and did not in fact disable the camera. Just because the application had control, and was not recording (as far as the user knew) anything, did not mean the camera was actually disabled. Android fixed this in a newer version by making the camera automatically release when the application that had control was no longer the application with which the user was interacting. This small change in how the OS functioned made obsolete all applications that handled camera "disabling" in this manner.

It was not until Android version 22 that Google began using administrative controls to control particular functions on the phone. In addition to allowing enterprise-level control of issued phones for network access, encryption, and some handling of data (contacts, security applications, and email accounts, for example), it also provided for a method of getting root-level control and disabling of the camera hardware in a software implementation. Once the camera is disabled at this level, then any photo capture is no longer possible, including screen captures.

Implementation is demonstrated in Chapter IV, but it is a straightforward approach that can be manipulated in a variety of methods, all of which function as an on/off switch for all photo access. For our prototype, we use an NFC tag as an input to start an app, which will disable the camera (along with other parts of the device). There are a couple of conditions for this to work. For any application to function with device administrator access, the user must give permission when it is first opened so that all accesses are granted. If the user does not give access, then the application will not function despite its presence on the smart device. Additionally, granting access does not actually cause any functionality to occur. With our application, the user must interact with an NFC tag. Once that has been done, the user cannot re-enable the camera functionality without scanning another, differently programmed NFC tag. That is to say, double scanning the tag that disables the camera will not actually turn the camera back on and the user does not have a button to subvert the design of the camera. Third, the camera is still disabled if the application is terminated by the user or if it were to crash. This is not the case for all items that are disabled, but is especially important to note when it is possible to keep the user from reinstating an access and subverting our application.

2. Microphone Considerations

A concern with microphones is the ability to record conversations that may be anything from official information up to top secret, compartmentalized information. Training several years ago focused on miniaturized digital recorders that could be hidden in a pocket or taped to a body. We were warned to be on the lookout for such devices and it was easy to raise a flag on spotting one. Today with a cellular phone in the hands of at least 90 percent of Americans and desensitization to them, we think less about what capabilities are on board.[215]

The microphone presents a unique opportunity since a user can start recording and drop the phone in a pocket where it can record unnoticed. It can also provide a direct line out to another location if a phone call is connected, allowing someone else to listen to the discussion occurring at a location, record remotely, or even accidentally expose classified details to an unknowing call recipient. This is mentioned because it is possible to have a phone in auto answer mode for driving or motorcycle riding with Bluetooth devices or just bump the answer button but never actually realize that someone has called. If accidentally answered or if auto answer is enabled, anyone that calls will hear what is going on and be difficult to detect until the physical device is looked at.

[215] Pew Research Center, “Mobile Technology Fact Sheet.”

A final reason to turn off the ability to use a microphone is because Android has shown vulnerabilities in the past which allowed hackers to access a microphone.[216] This allows for listening and recording of conversations in a situation where the phone's owner would have no idea what was happening. The likelihood of this is very low on a typical device, there is a trend among more savvy android users to "root" a cell phone. "Rooting" will be discussed later, but essentially it allows for more flexibility and disables some Android OS safeguards so that the user has more control. With this done, it also allows hackers easier access to phone features, specifically a microphone or camera, and would open the user to a higher potential for exploitation.[217]

Disabling the microphone requires access permission, but at a lower level of access than that of the camera. That is to say it does not require device administrator support. Chapter IV discusses how we approach disabling the mic. We utilize a toggle of the microphones mute feature. This method does not function at a level that would prevent it from being toggled back on if our application were to conflict with another application's attempt to unmute the mic. Note that another application would have to specifically access the unmute method to override our mute. The application is set up to show the current status of the microphone when it is started. When it performs that initial check, which is a Boolean check, it will know if the microphone is muted or not and will display the appropriate message in the bottom left side of the screen with the status of other components.

There are other ways to help mitigate this issue that should be considered in an application that is ready for release to sailors, however for this proof of concept we simply wanted to show that it could be done. In a polished application it would be smart to go ahead and address input volume as well as muting the mic. A programmer could set this to zero so that if the mic is not muted, then it is not registering a pick up of sound

[216] Ms. Smith, "Black Hat: It's Not ''Tricky' for Hackers to Turn Your Phone into a SpyPhone," Network World, August 1, 2013, http://www.networkworld.com/article/2225081/microsoft-subnet/black-hat--it-s-not--tricky--for-hackers-to-turn-your-phone-into-a-spyphone.html. Ms. Smith is a pseudonym used by the author.

[217] Veo Zhang, "Hacking Team RCSAndroid Spying Tool Listens to Calls; Roots Devices to Get In," *TrendLabs Security Intelligence Blog*, July 21, 2015, http://blog.trendmicro.com/trendlabs-security-intelligence/hacking-team-rcsandroid-spying-tool-listens-to-calls-roots-devices-to-get-in/.

until the user or some other application bumps it back up. Taking a multi-step approach to the microphone must be considered since actually disabling it like the camera is currently not available.

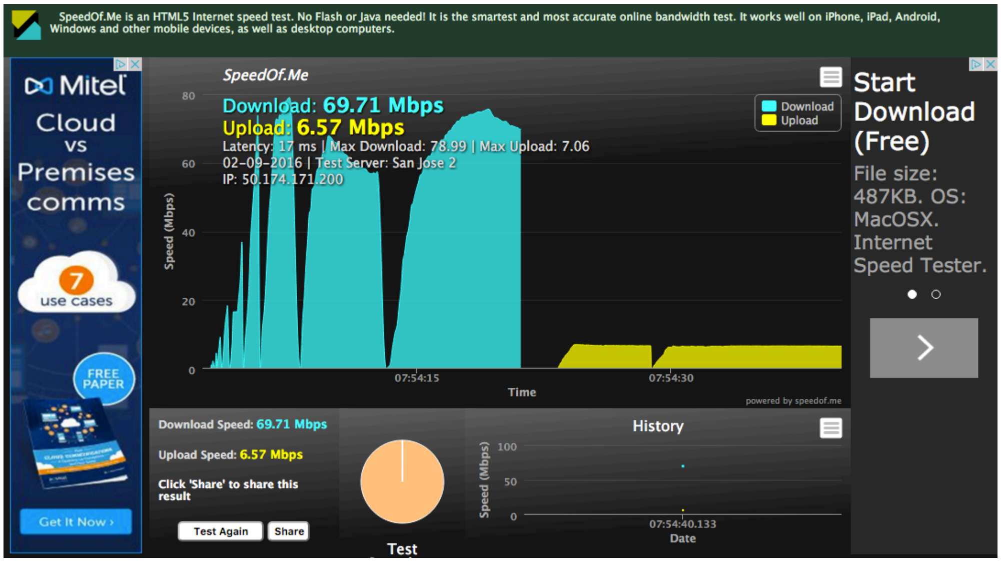
3. Wi-Fi Considerations

Wi-Fi technologies have existed for years and over time their communication reliability, range, and data transfer speed has improved. These technologies have opened up a whole range of new uses on smart phones. It is no longer just a simple tool to enable higher data rates to browse the Internet. Now, smart devices can be linked through local Wi-Fi connections. Remote controlling can be implemented with onboard applications for everything from communications with other smart devices to light bulbs and data can travel with high reliability between multiple devices on the same network.

For smartphones with a reliable Wi-Fi connection it is not unreasonable to see transfer speeds comparable to (or even in excess of) that of a computer. There was a time when accessing the Internet, even via Wi-Fi, on a mobile device was considerably slower than on a full computer but that difference has diminished significantly. Onboard smartphone Wi-Fi is no longer limited to the older, more miniaturized technologies, and can operate at the newest standards available for 802.11 in many cases.

We performed two separate speed tests on the same wireless network on both a personally owned 2011 MacBook Pro and an iPhone 6+ (Figures 12 through 15). The tests were done using Ookla’s speedtest site (www.speedtest.net) and Speedofme’s site (www.speedof.me). These are considered to be two of the more reliable online bandwidth tests with the differences in each not being a critical factor for the scope of this thesis. Rather, this demonstration shows with the two tests that in both cases a modern smartphone had faster data transfer speeds than a reasonably modern laptop. The MacBook is running a current version of El Capitan, the most modern operating system, has 16GB of onboard RAM, and a 2.3 GHz Intel I5 processor (Dual Core). The iPhone 6+ is running the most current version of iOS 9, has 1GB of onboard RAM and a 1.4 GHz Typhoon processor (Dual Core).

Figure 12. Screen shot of www.speedof.me in-browser speed test for MacBook Pro



SpeedOf.Me is an HTML5 Internet speed test. No Flash or Java needed! It is the smartest and most accurate online bandwidth test. It works well on iPhone, iPad, Android, Windows and other mobile devices, as well as desktop computers.

SpeedOf.Me

Download: 69.71 Mbps
Upload: 6.57 Mbps
Latency: 17 ms | Max Download: 78.99 | Max Upload: 7.06
02-09-2016 | Test Server: San Jose 2
IP: 50.174.171.200

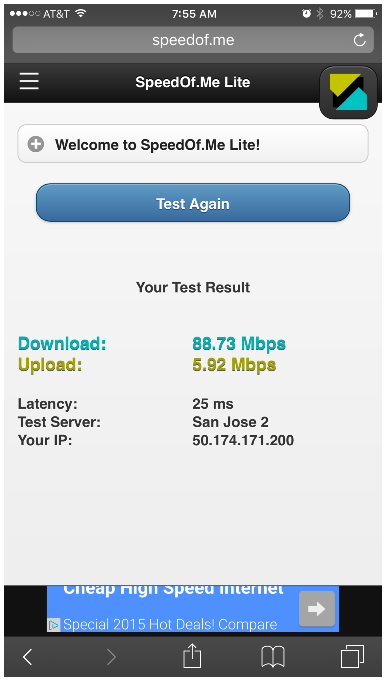
Download Speed: 69.71 Mbps

Upload Speed: 6.57 Mbps

Click 'Share' to share this result

Using www.speedof.me to perform a speed test, the MacBook Pro achieved a maximum download speed of 69.71 Mbps across multiple size file samples. Upload speeds peaked at 6.57 Mbps.

Figure 13. Screen shot of www.speedof.me in browser speed test, mobile version, for iPhone 6+



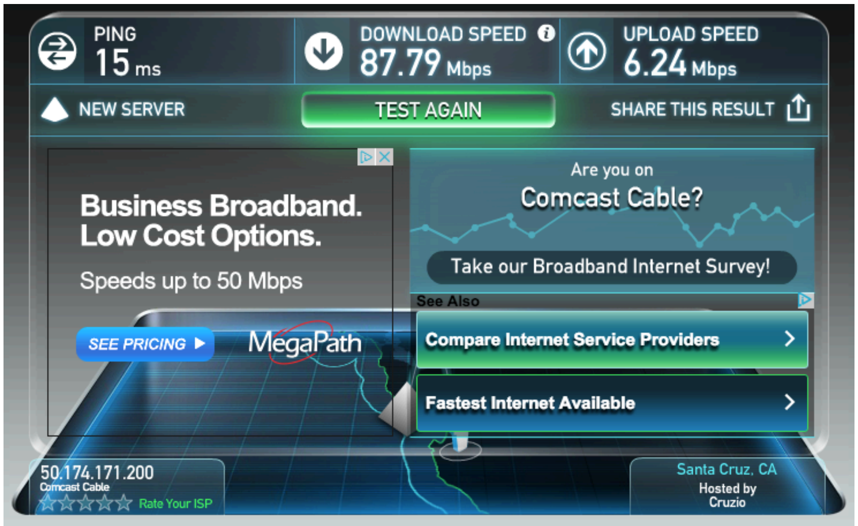
Welcome to SpeedOf.Me Lite!

Your Test Result

Download:	88.73 Mbps
Upload:	5.92 Mbps
Latency:	25 ms
Test Server:	San Jose 2
Your IP:	50.174.171.200

Accessing www.speedof.me on the iPhone 6+, maximum download speeds were 88.73 Mbps and uploads peaked at 5.92 Mbps.

Figure 14. Screen shot of Ookla's www.speedtest.net in-browser speed test for MacBook Pro



PING 15 ms

DOWNLOAD SPEED 87.79 Mbps

UPLOAD SPEED 6.24 Mbps

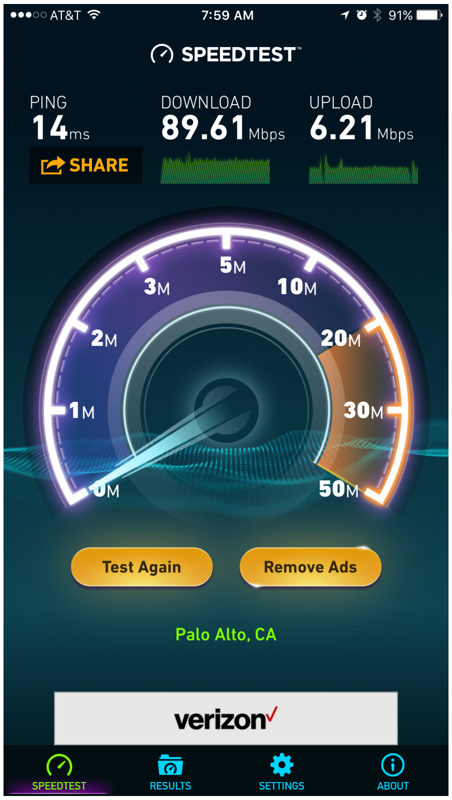
50.174.171.200
Comcast Cable

Santa Cruz, CA
Hosted by Cruzio

NEW SERVER
TEST AGAIN
SHARE THIS RESULT
Business Broadband. Low Cost Options.
Speeds up to 50 Mbps
SEE PRICING
MegaPath
Are you on
Comcast Cable?
Take our Broadband Internet Survey!
See Also
Compare Internet Service Providers
Fastest Internet Available
Rate Your ISP

Using www.speedtest.net to perform a speed test the MacBook Pro achieved a maximum download speed of 87.79 Mbps. Upload speeds peaked at 6.24 Mbps.

Figure 15. Screen shot of Ookla's speed test using the iPhone application



PING 14 ms

DOWNLOAD 89.61 Mbps

UPLOAD 6.21 Mbps

Palo Alto, CA

Using Ookla's iPhone application to perform a speed test on the iPhone 6+ the maximum download speed was 89.61 Mbps with uploads peaking at 6.21 Mbps

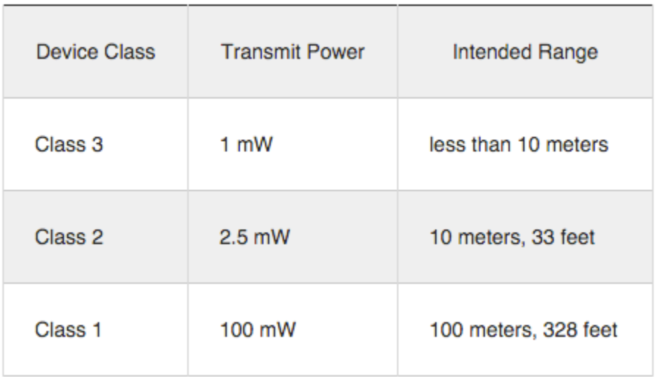
Disabling Wi-Fi in our application is handled in a fashion similar to the microphone. What is interesting, however, is that the state is not saved so if the application is disabled, crashes, or is forced to restart, then the NFC tag must be rescanned to disable Wi-Fi again. While the application is running, Wi-Fi will remain off and cannot be re-enabled in the user's settings. As mentioned in the general application discussion, a developer would have to implement more robust implementations to restart and save device settings in the event of a restart.

4. Bluetooth Considerations

Bluetooth, much like NFC, creates a way for two devices to exchange data using RF transmissions. It is traditionally associated with earpieces to communicate during phone calls, speaker systems to listen to music, cameras to view a video feed, or a method to pair and control many other devices like smart home components or remote controlled drones. It also allows for pairing for input/output devices like keyboards and mice, eases transferring files, and can automatically configure Wi-Fi settings for a new network. It is a very flexible, powerful communication technology that is now included on virtually every smart phone and many forms of tablets.

Bluetooth's frequency range and power allows for much greater range than NFC, and its ability to pair devices makes it especially appealing to users of the technology. It is the range of Bluetooth that also presents particular concern (Figure 16). Bluetooth can carry any data over a connection between paired devices and is only limited by what the application using it is requesting. A phone can send back a host of information that is useful to a user and the typical thought that it is strictly a short-range communication is misleading. Depending on the class of Bluetooth on a device data can be sent and received up to 100 meters away in unobstructed environments.

Figure 16. Graphic showing expected Bluetooth RF ranges



Device Class	Transmit Power	Intended Range
Class 3	1 mW	less than 10 meters
Class 2	2.5 mW	10 meters, 33 feet
Class 1	100 mW	100 meters, 328 feet

Source: Joshua Wright, “Dispelling Common Bluetooth Misconceptions,” Sans Technology Institute, Security Laboratory, accessed February 1, 2016, https://www.sans.edu/research/security-laboratory/article/bluetooth.

The many exploits and vulnerabilities associated with Bluetooth technology could provide enough information for a paper that would rival the length of this thesis and is readily available on the Internet and in tech journals. For this reason and the potential range, disabling Bluetooth communication on a device should be a main feature of any lockdown application. In our implementation shutting down Bluetooth functions in much the same way we shut down Wi-Fi. It will be off as long as the application is running but it can be subverted by disabling the application or if the application crashes. Much like Wi-Fi, it would be advisable to save the current state in a file within the application so that Bluetooth capability status can be controlled if a restart occurs.

5. Messaging Considerations

Text messaging on Android is handled via a couple of possible text exchange manners. The default application is called Google Hangouts and it allows for sending of traditional short message service (SMS) and multi-media service (MMS) messages to other phone numbers, google hangout contacts, or email addresses. Google Hangouts will send the data through the cellular network when no Wi-Fi is present, or will chose to send messages over Wi-Fi if available and the message recipient is an email address, a user logged into a browser with the Hangouts plug in, or a phone number that also uses

Hangouts.[218] Looking only at Hangouts or considering basic messaging does not capture the actual picture of how users now communicate with each other.

Hangouts is a robust messaging application, but it is not the only way that users on Android send text messages. Messaging service apps have become very popular and the features offered now include much more than just text transfer. Many, such as Kik or Facebook Messenger, offer the ability to send photos and videos. Some, such as Skype, even offer direct connections to video chat with multiple people in real time.

Messaging and data exchange over cellular networks has also opened smart devices to greater exposure to traditional attacks. Clicking a link in a text message can lead to a site that executes malicious code or asks permission to install a malicious third party application. Smart devices that automatically download the contents of a multimedia message can also download malware without any user interaction and leave it open to being hacked or having its data exploited.[219] Users should absolutely be aware of what they are clicking and who it is coming from, but it is also important that the device be updated with all vulnerability patches. Automatic downloads of data should be blocked, and users should be trained on why this creates greater vulnerabilities in their personal devices. This is not an easy consideration for Android, but is necessary for it to function within a DOD environment.

The final consideration for text messaging is that this feature, whether on the native app or in some third party app, still provides a means to record data and transport it. Even if a message is opened and never sent, it can be used as a note pad to record observations or copy text from classified documents. The information in that text will stay in the app, and can be added to as long as the application is open. For this reason, future versions of our application would have to address this problem.

[218] Google Support, "Get Started with Hangouts," Google Support, accessed February 10, 2016, https://support.google.com/hangouts/answer/2944865?hl=en.

[219] Dan Goodin, "950 Million Android Phones Can Be Hijacked by Malicious Text Messages," Arstechnica, July 27, 2015, http://arstechnica.com/security/2015/07/950-million-android-phones-can-be-hijacked-by-malicious-text-messages/.

There are a couple of ways this could be done, but neither would avoid possibly being patched out of future versions of Android. Currently, one could implement a custom keyboard that does not have any letters on it. While our app is running, a user could still receive a text message but would be unable to type a response. The native keyboard would be suspended in exchange for our app's keyboard. In this way, a user could still receive urgent messages while in cellular coverage, but would not run the risk of transferring or recording classified or official use information. The other to mitigate the risk would to be to have our application kill any applications that are not authorized. This would require a more robust coding package, and would require a white list of authorized applications. If an application was not specifically authorized by the security lockdown app, it would be killed as soon as it was started. Both functions would work, but would require more programming than was covered in our example and was not implemented in this application.

6. Cellular Data Considerations

A final area that should be looked at is the transfer of cellular data to and from a mobile device. We attempt to shut this down and were successful with older versions of Android, but as discussed later we are unable to do so with the current implementations of Android. We mention it here because a compromised device could still send information off without a user's knowledge and this is a significant consideration.

It is desirable that when the application is activated all cellular data exchange is disabled. Having this ability stops applications on the phone from transferring data to other parties. This will essentially turn a smart device, with all other security implementations in place, into a cellular handset much like the Nokia phones that were popular in the late 1990s. The cell phone could then send and receive traditional SMS/ MMS messages and make phone calls. However, users cannot browse the web or post status updates to social media. The phone, without the ability to take photos, no ability to record with a microphone, with no access to Wi-Fi or Bluetooth, and with no cellular data capability will have been locked down to a level where considerations are more manageable and building out additional BYOD features becomes a possibility.

C. ANDROID OS CONSIDERATIONS

Android's OS controls greater than 80 percent of the smartphones currently in use around the world. Additionally, it is widely used in tablets. Its flexibility and customizability for manufacturers has allowed a wide market distribution. It is used on both low-end, bargain-priced tablets as well as high-end devices from manufacturers like Samsung and Nexus. The ability for manufacturers to use it as they see fit and load it exactly as they would like also creates significant difficulty in rolling out patches for vulnerabilities. Google places the onus on manufacturers to roll out updates as they are made available. This compared with iOS, which controls and encourages updates, means that there is a much greater amount of fragmentation in the Android OS market.[220] Manufacturers must put forth the effort to provide device OS updates and upgrades to their users. Those of low-end smart device offerings have little incentive to push updates in a timely manner to older versions of Android and often devices are often left with unpatched vulnerabilities.[221]

Looking beyond the difficulties associated with upgrading/updating the Android OS sitting on most smart-devices, there are many features, which are employed that are very useful and successful and keeping devices safe. The biggest obstacle to the security mechanisms on the device is the end user. While taking an operating systems class we both worked with Jerel Yam (another student) to specifically explore how typical malware can exploit Android vulnerabilities. What we found was that it was Android's flexibility coupled with the user's lack of attention that created a dangerous environment for the operating system. Android has implemented methods such as sandboxing, using unique user/namespace instances, and limiting resource access and communication between applications. Mandatory Access Control (MAC) lists are enforced and permissions associated with each process are not communicated to other processes. These

[220] "Why Hasn't My Android Phone Updated Yet?" MakeUseOf, accessed February 10, 2016, http://www.makeuseof.com/tag/why-hasnt-android-phone-updated/.

[221] Ibid.

are indicative of significant efforts to provide the safest, most stable OS to the users while maintaining a high degree of customizability for manufacturers and users alike.[222]

If BYOD is to eventually be an option in any DOD environment, then policy must exist to cover the gaps that do exist in Android's OS (as well as that in iOS). Users have the ability to root a device, which is a way of saying opening up all root access and is similar to jailbreaking in iOS. This degrades the security features on the phone and gives malware and applications a method to manipulate the device at a level normally not permitted. In addition to rooting a device to change security settings, Android also allows users to explore applications outside of its Play Store. Third party application providers do not always have the same requirements for security screening before posting an application for download. When users step outside of the main, reputable providers they open themselves up for greater malware infection. Policy must account for this and implementation must have a way to verify that all OS-specific security features are operating and have the most current updates. While this can create difficulty given Androids difficulty to update/upgrade, it will ensure that as the device operates it is not performing unexpected or unauthorized execution of code.

1. Android Development

The second broad effort of our document research related to the development of an Android application discussed further in Chapter IV. The review has technological implications, which helps it fit more appropriately in Chapter III. We begin our research by attempting to develop a basic application in Android. This effort started in winter 2014. This date is important because at the time, Eclipse was the principle IDE for developing Android applications. Meanwhile, Google was in the process of enriching and debugging their own standalone IDE Android Studio. In the winter of 2014–15, Android Studio was heavily criticized as unstable and broadly unusable for application development. We focused our efforts on utilization of *Murach's Android Programming* methods for the remainder of 2015. With numerous smaller applications built in Eclipse,

[222] Travis Miller, Jerel Yam, and Liam Dorney, "An Examination of Malware Interactions in the Android OS" (class paper, CS3070, Naval Postgraduate School, March 2015).

we started to realize that updates to the IDE and builds were harder and harder to install and run successfully. This drove our migration to Android Studio in September of 2015. The transition was much easier than expected and accomplished using *The Big Nerd Ranch Guide to Android Programming*. Overlooking the obvious naming convention issue, we found this book was highly recommended and received excellent reviews from online android programming communities and stores.

2. Murach's Android Programming

We began our research into Android development with Murach's Android Programming during the winter of 2014–2015. As stated above, Eclipse was the IDE most widely used in Android application development in 2014 and prior. Eclipse was the backbone of Murach's book. The amount of time trying to get Eclipse up and running and then using it to programmatically access and control the features we sought to control, such as camera, microphone, Wi-Fi, were excessive. Numerous hours were spent solving Eclipse for Android issues rather than the device features themselves, such as numerous emulator and device build fails due to unsupported device features. Furthermore, the versions of Android we were able to access tended to be two to three builds behind current versions of Android OS. This made online support for the programs and techniques in Murach less appealing from the commencement of a project. We offer this information not as critique of the book itself, but rather as an important lesson learned through our research: the absolute necessity of staying current with IDEs and associated reference texts for the operating system in question. This rule of thumb is even more important in the world of mobile application development due to rapid changes in mobile technology. Significant time was spent in Murach's book utilizing techniques and build styles that were less relevant to the current Android OS, but no longer supported by software development kits (SDK), or better implemented in Android Studio. After numerous attempts to get our training applications up and running, and looking at the most current methods of programming in Android we made the decision to transition to Android Studio.[223]

[223] Joel Murach, *Murach's Android Programming* (Fresno, CA: Mike Murach & Associates, 2013).

3. Android Programming: The Big Nerd Ranch Guide

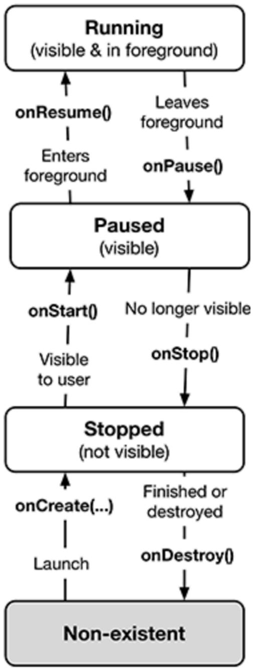
The *Big Nerd Ranch Guide* (BNRG) proved to be an invaluable asset across the board in the early stages of our Android Studio (AS) experience. The book itself and the accompanying update website provided thorough instruction on installation and basic operations in AS. The first noticeable difference in IDEs presented in BNRG is the AS project builder. BNRG discusses what became a pivotal point in our research—the AS Android Target Device selection.[224] We discuss this topic and its impact on our research further in Chapters IV and V.

BNRG provided another key research point in its coverage of the activity life cycle. Critical to Android programming BNRG provide the following explanation: "During this life cycle, an activity transitions between three states: running, paused, and stopped."[225] The book goes further and provides the graphic in Figure 17. This helped identify the points of potential intercept for our applications listeners when trying to control instance of camera, microphone, Wi-Fi, data etc. On numerous occasions, we visited this graphic to plan our attack on various activities further discussed in Chapter IV and V.

[224] Bill Phillips, and Brian Hardy, *Android Programming: The Big Nerd Ranch Guide*, Vol. 2 (Atlanta, GA: Big Nerd Ranch, 2015).

[225] Ibid.

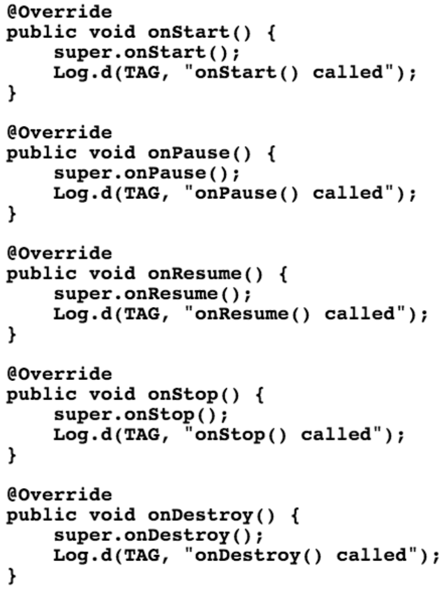
Figure 17. The activity life cycle



Source: Bill Phillips, and Brian Hardy, *Android Programming: The Big Nerd Ranch Guide*, Vol. 2 (Atlanta, GA: Big Nerd Ranch, 2015).

BNRG took us one step deeper and walked through the logging of the activity life cycle to eavesdrop on every method call within each class. Again, this technique was essential for identifying locations for listeners on OS attempts to access various features we sought to lock. An example of the BNRG code used to log activity calls is presented in Figure 18.

Figure 18. Logging activity calls at every stage of the activity life cycle



```
@Override
public void onStart() {
    super.onStart();
    Log.d(TAG, "onStart() called");
}

@Override
public void onPause() {
    super.onPause();
    Log.d(TAG, "onPause() called");
}

@Override
public void onResume() {
    super.onResume();
    Log.d(TAG, "onResume() called");
}

@Override
public void onStop() {
    super.onStop();
    Log.d(TAG, "onStop() called");
}

@Override
public void onDestroy() {
    super.onDestroy();
    Log.d(TAG, "onDestroy() called");
}
```

Source: Phillips, and Hardy, *Android Programming*.

Another key learning point seen in Figure 18 was the application of a Java code strategy invoking super on the class of each activity. This invoked the parent class that gave access to any instance of that class call or construction.

To avoid excessive duplication, we cite BNRG often in later chapters as we use the tools and lessons discovered therein. The essence of this discussion is to point out how vital current resources are when trying to develop mobile applications where change occurs and new technology supports learning. BNRG was vital in our understanding of intents, listeners, and activities as they relate to the features we tried to control from our application.

4. Google Android Development

This source provided the backbone of our application development. We save the more technical aspects of our research for Chapters IV and V, but for the purposes of identifying pivotal Android development access points we identify some key learning onramp locations. First among these onramps was the foundation of GAD that is based on programing in the Android Studio (AS) IDE. We initially struggled as discussed above when trying to code in Eclipse as every Google document or webpage we tried to access referenced features or tools only available in AS. This was the driving factor utilizing AS as our principal IDE for application development. Once we embraced AS, GAD opened a plethora of development assets that proved essential for our research. The remaining onramps for our research are based on the GAD structure loosely defined in two categories, design and develop, and are discussed below. We discuss the design specifics of our application in Chapters IV and V, but provide the basic elements of GAD design principles here to ensure this and future research considers Googles principles when developing applications.[226]

a. Developers Design

Google sets standards for the design of applications running on the Android OS. Google enumerates several design features for optimization of presentation and user experiences. Any user of an Android device knows the wide disparity in the design layout of the hundreds of thousands of applications available via the Google Play store. In what could be compared to interior design guidelines, Google has created a standard for the user experience while interacting with Android applications. The Developers Material Design (DMD) page is the virtual guidebook for these standards and breaks these guidelines down into the following groups:

(1) Animation

Animation covers motion, responsive interaction, transitions, and details. Motion covers the elements of object movement throughout the application layout. Spatial relationships between elements within the design are covered here. Google has gone so far as to study the amount of attention user pay to objects as they enter and exit the space. An example given is the entrance and exit of objects, menus and pictures, where by the recommended setting calls for increase in speed when object exit and the opposite when

[226] Developers, “Get Started with Android Studio,” Developers, accessed February 1, 2016, http://developer.android.com/develop/index.html.

entering the frame.[227] The responsive interaction section discusses surface reaction and tactile responses. Recommendations include material popups should expand out of the point of input, as opposed to random spots on the interface.[228] Radial action provides further clarity to the user, doing so here by adding clarity for the user through a visual reaction to their input.[229] Finally, worth noting in the interaction subcategory is the note that an input should provide action such as a transition that is visually connected to the point of input, specifically calling out that the input should provide a direct transition to related information.

(2) Components

Components is the compendium of parts design specifications that make up the application. From the three types of button (floating action, raised, and flat) to menu specification for optimized experiences the Design Components pages has a design recommended specification, examples of those specifications, and samples for download. Google design built into coded layouts and expected behaviors was especially helpful in our research. Simple UI features often overlooked by developers can be preprogrammed into an application by utilizing GAD's design samples for use. An example is the built-in behavior of menus so that the menu is positioned over the emitting elements.[230]

(3) Layout

The most useful section in the design elements in our research is the layout principles section. The Metrics and Keylines sections give representative examples of virtually every application layout available with spacing and orientation design elements for optimized viewing (Figure 19).

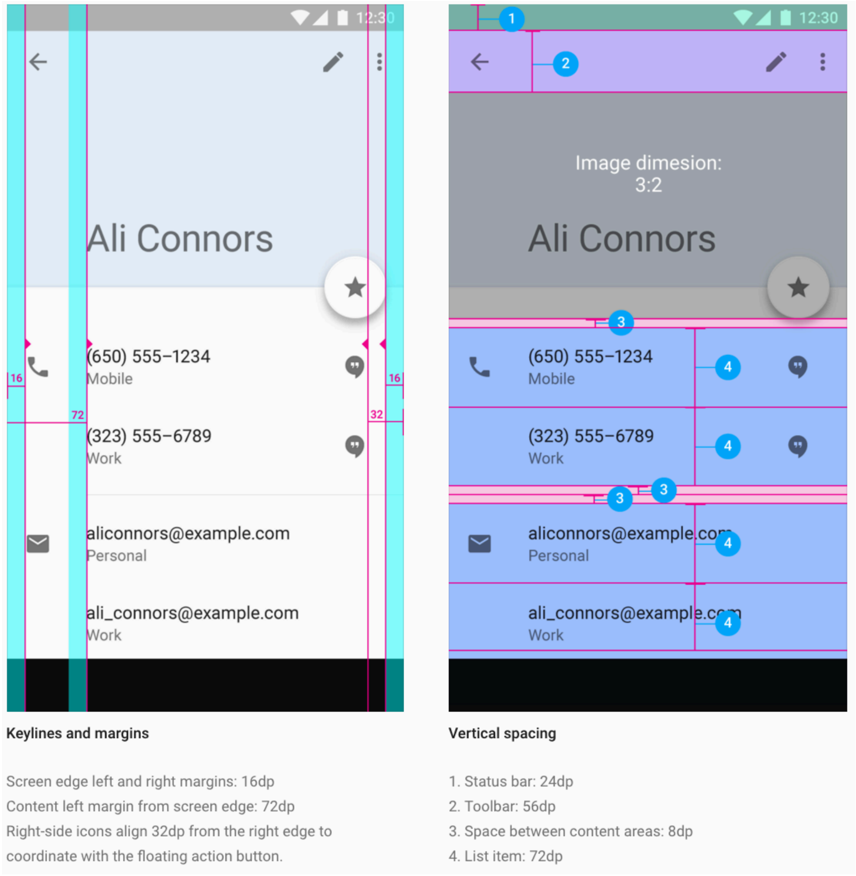
[227] Google, "Authentic Motion—Animation—Google Design Guidelines," Google, accessed February 1, 2016, https://www.google.com/design/spec/animation/authentic-motion.html#.

[228] Ibid.

[229] Ibid.

[230] Google, "Menus—Components—Google Design Guidelines," Google, accessed February 1, 2016, https://www.google.com/design/spec/components/menus.html#menus-behavior.

Figure 19. Example metric and keyline layout of phone application



Keylines and margins

Screen edge left and right margins: 16dp
Content left margin from screen edge: 72dp
Right-side icons align 32dp from the right edge to coordinate with the floating action button.

Vertical spacing

1. Status bar: 24dp
2. Toolbar: 56dp
3. Space between content areas: 8dp
4. List item: 72dp

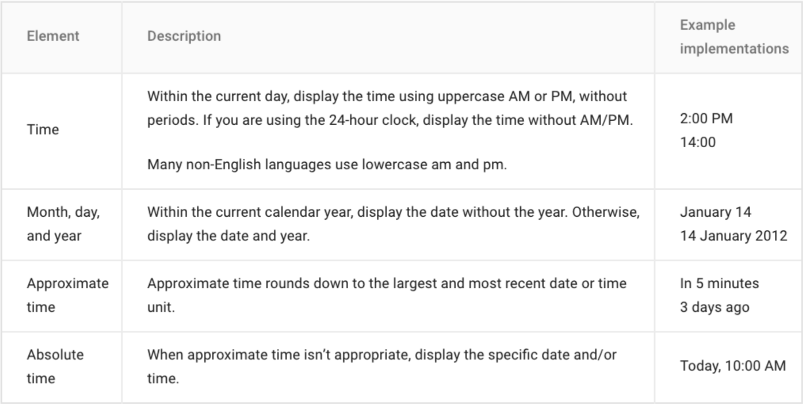
Google, "Metrics & Keylines—Layout—Google Design Guidelines," Google, accessed February 2, 2016, https://www.google.com/design/spec/layout/metrics-keylines.html#metrics-keylines-keylines-spacing.

As a point of reference, the layout of our application was based on information and specifications gleaned from the Design layout section.

(4) Patterns

Patterns contains the general styling parameters of typical situations that arise in the most common applications. An example is the date and time parameters shown in Figure 20. This is one of several tables on the date and time patterns page. We offer this example to demonstrate the thoroughness and depth of the Google design team specification guidance that many developers violate. To extent the point there are seventeen specific patterns that Google has identified as common and offers standardized solutions for them.

Figure 20. Date and time parameters and implementation



Element	Description	Example implementations
Time	Within the current day, display the time using uppercase AM or PM, without periods. If you are using the 24-hour clock, display the time without AM/PM. Many non-English languages use lowercase am and pm.	2:00 PM 14:00
Month, day, and year	Within the current calendar year, display the date without the year. Otherwise, display the date and year.	January 14 14 January 2012
Approximate time	Approximate time rounds down to the largest and most recent date or time unit.	In 5 minutes 3 days ago
Absolute time	When approximate time isn't appropriate, display the specific date and/or time.	Today, 10:00 AM

Google, "Data Formats—Patterns—Google Design Guidelines," Google accessed February 4, 2016, https://www.google.com/design/spec/patterns/data-formats.html#data-formats-date-time.

With respect to patterns, our research implemented the patterns recommended in the Notifications Patterns page. The exact notification patterns used is discussed further in Chapters IV and V.

(5) Style

Style makes up the largest of the design elements google has provided specifications for. In the Style Color section, Google provides thousands of color palettes available to the developer with pairing recommendations in the Color Scheme section. Further amplification on style is given for text, with follow on text and background color guidance. Style Icons is the next section where specification for design and implementation of the application icon is thoroughly laid out. From the five parts of a icons anatomy (finish, material background, material foreground, color, shadow) through to the design principles associated with each level, Google has completely specified icon development and use for optimal user experience. The Imagery Style section provides guidance on the use of Images in an application. This is especially useful for our application as the guidance is was used to format and position the imagery used, which is discussed further in Chapter V.

(6) Usability

According to Google, usability should be one of the foremost design principles an application developer adheres to. With the goal of maximum usability consideration must be given to the ideas of quick, effective, and efficient navigation for every user.[231] Google again provides examples and samples for download and implementation in design, and further encourages its specifications in those samples. Usability was one of the key sections we used in the design of our application. We attempt to adhere to the principle of complete access discussed in this section and will elaborate in Chapter V.

b. Developers Develop

The meat of our research is conducted within the Develop pages. Getting used to AS was difficult and we often could not rely solely on BNRG for executing tasks with in the IDE. The homepage offers extensive guidance on installation a setup of AS on any OS. This is indispensable in the onset of application development as early installation was successful but failed after use due to incorrect installation of AS's SDK manager. The six portals available on the Develop page branch the user off quickly and effectively based owned and experience and are discussed further below.

(1) Training

This is home too much of our early research as we struggle through the initial stages of Android development. Thankfully, the training portal offers a "building Your First App" session that walks the user through set up of a simple text display and triggering the display with a button. Based on the interactive nature of the majority of the Android applications available in the Play Store, Google emphasizes the importance of buttons and their use through this tutorial.[232] With the thorough coverage of intents in the

[231] Google, "Accessibility—Usability—Google Design Guidelines," Google, accessed February 4, 2016, https://www.google.com/design/spec/usability/accessibility.html#accessibility-types.

[232] Developers, "Starting Another Activity," Developers, accessed February 4, 2016, http://developer.android.com/training/basics/firstapp/starting-activity.html.

training app we are able to execute some of the more vital functions in our security application, and we highly recommend anyone attempting to start Android programming begin with these guides.

(2) API Guides

The API guides provide a more advanced framework for development in Android. Here is where we learned the four tenants of Androids security sandbox operating environment. First each application is considered an individual user in the multi-user operating system, whereby multiple apps run as multiple individual users (Linux multi user system).[233] Second, the OS assigns unique user identification to each application, and that identification number is known only to the operating system.[234] Associated with that number is a set of system permissions designated by the developer, but governed by the operating system.[235] For example, an application cannot grant critical system permissions without the user authorizing such permission, after which the ID number is assigned those approved permissions. Third, each application's code is run on an independent virtual machine (VM) separate from the other VMs for other running applications.[236] Fourth, all applications run in their own Linux process, starting when onCreate is called, and shutting down with onDestroy or when the Android OS must recover memory for system functions and or other applications processes.[237]

This section is also pivotal in our understanding of the four essential components of an application. We list and discuss them here as a reference for future research. (1) Activities are exactly what the word describes. These are the things that get the work done within the application, and each application could have several or only one of them depending on the complexity of the application. The application given by Google demonstrates an activity best as used in an email application. One activity might retrieve

[233] Developers, "Application Fundamentals," Developers, accessed February 4, 2016, http://developer.android.com/guide/components/fundamentals.html.

[234] Ibid.

[235] Ibid.

[236] Ibid.

[237] Ibid.

new mail, another might allow the user to compose new mail, and yet another might take care of deleting mail.[238] (2) Services perform the long-term tasks that do not involve user interface such as playing music on a playlist.[239] These services are initiated by an activity and stopped by an activity. (3) Content Provider provides data management between applications in any location that the application has access to store data.[240] (4) Broadcast Receivers respond to broadcasts that are made system wide such as an announcement that the battery is low, the screen is turned off, or an application has accessed the camera.[241] These responses can initiate activities, cause something to happen with in the OS, or limits other work functions in the system. Understanding these four components proved crucial to our research in learning how to target the application components we were looking to control. Explained here by Google and with numerous links to related information, formatting, use, and training modules made this section a go-to-part of our research.

(3) References

The technical library of the Android Developer experience, section was accessed several times. In fact, the reference library contains the entire bank of packages, classes, subclasses, methods within each class, every broadcast receiver available to the programmer. An example of a MediaController is provided in Figure 21. Notice that this image is only an introduction to the class. Several pages of data on the class is available and hyperlinked to further explanations and usages. For example, all the public methods are available, with the associated input parameters and data types. It goes without saying searching the developer references for class and method uses occurred regularly and was a great source when issues with code arose.

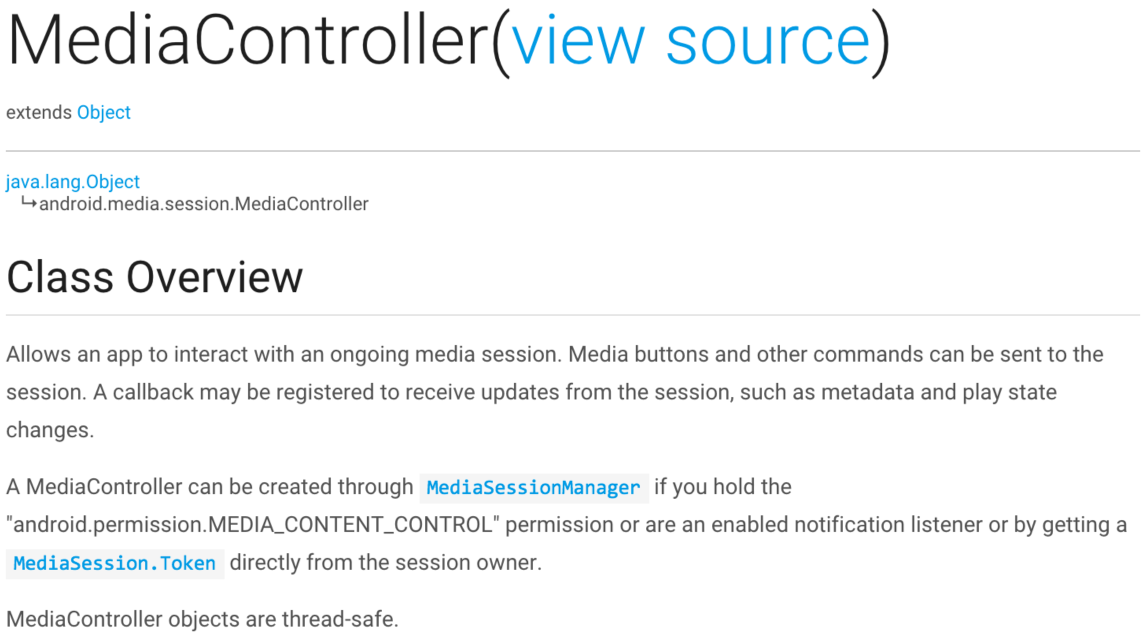
[238] Ibid.

[239] Ibid.

[240] Ibid.

[241] Ibid.

Figure 21. MediaController class in Google Developer



MediaController(view source)

extends Object

java.lang.Object
↳android.media.session.MediaController

Class Overview

Allows an app to interact with an ongoing media session. Media buttons and other commands can be sent to the session. A callback may be registered to receive updates from the session, such as metadata and play state changes.

A MediaController can be created through `MediaSessionManager` if you hold the "android.permission.MEDIA_CONTENT_CONTROL" permission or are an enabled notification listener or by getting a `MediaSession.Token` directly from the session owner.

MediaController objects are thread-safe.

Developer, "MediaController," accessed February 1, 2016, http://developer.android.com/reference/android/media/session/MediaController.html.

(4) Tools

Tools provides the AS interfaces for basic through advanced development. Starting with complete system tours of the Android Studio IDE and including plugins for Gradle and Manifest Merging.

(5) Samples

Samples contains hundreds of bare bones application and API samples. These are useful within applications and have been provided by Google for use in development. The samples themselves serve to help standardize Android applications by providing a base to build from. Often times the samples require build out of components. We utilized several samples applicable to our application, specifically the device police controller API, which we discuss further in Chapters IV and V.

5. StackOverflow

StackOverflow[242] was invaluable in our build process. We set out trying to take control of very specific features of mobile devices, namely camera, microphone, Wi-Fi, Bluetooth, and Data. Without the numerous blog entries and code samples posted by the stackOverflow community we would not have been able to complete our work. The visits to stackOverflow were made most often for help with the most complex development questions and when trying to access root functions within the Android OS.

6. Android Weekly

Android Weekly[243] provided essential guidance and in depth examples that enabled working through some of the more advanced parts of our Android coding. Article number 183 stands out as a good example, as it included coverage of the use of permissions, which was a significant part of our coding experimentation. At one point the Android Weekly article on permissions saved us from what we though would be a complete rewrite of a class, when basic permissions modifications to the manifest file were all that were needed.

D. EXAMPLE SCENARIOS

We see our application fitting in as an initial offering to meet a larger goal with DISA and DOD. Enterprise offerings and solutions exist, but do nothing to address the users reporting to ships and installations with smart devices in their possession. By showing that the dangerous-to-security features can be disabled we hope that we can improve the probability that these powerful devices will be used to enhance the environment in which we operate. The example scenario below goes beyond what our application demonstrates, but also shows why it could help the DOD, ship's COs, and end users to have greater access to data on personal devices.

[242] Stack Exchange Inc., "Bounty 'Android' Questions," Stack Exchange Inc., accessed November 14, 2015, http://stackoverflow.com/questions/tagged/android?sort=featured.

[243] Gyuri Grell, Martin Gauer, and Sebastian Deutsch, "Android Weekly," *Android Weekly*, 2016, http://androidweekly.net/.

1. Preparation on the Ship

Ships' captains are tasked with protecting their ship from all forms of harm and ensuring that security is maintained at all levels. When a sailor reports to a ship with a cellular phone, tablet, laptop, and other electronic devices the policy should be written so that the sailor has some ability to use these devices while maintaining official and classified data's integrity. It is up to the security manager on each ship to work with the captain and, within DOD policy, set up standards that best reflect the captain's level of comfort and conform to best security practice standards.

In this scenario, that conversation has taken place. The security manager understands that the captain wants all smart phones carried on sailors to be as locked down as possible. The captain has also made it clear that despite this security feature, phones and mobile smart devices are still not to be carried into secure spaces and he does not want to see them in combat or radio. The security manager can now sit down at his desk and start programming the NFC tags that have been provided by the Navy. The first two tags are those that will turn the application on and lockdown all of the above features and the second tag will turn them back on and disable the application. After writing data to the tags, the security manager will verify that they are active by placing a smart device in close proximity and observing the device. If all is well, the application will open and a message listing which components have been disabled will be displayed on the screen. From here, the security manager moves to verify that these functions are in fact disabled by trying to access them. For example, attempting to take a screen shot and accessing the camera should throw an exception that those features are not available. Checking the phone's photo album would show that no new photos had been added. At this time, the security manager will place the two tags at the quarterdeck in preparation of sailors arriving on the ship and locking down their phone.

All of this preparation can be done on a ship's computer with an NFC encoding program in place. Rules will have been set up so that the security manager will simply click those features that need to be locked, and when the NFC tag is written it will be locked so that the data on the tag cannot be changed. To update the tag would no longer be possible and would require a rewrite to the change in policy. This would only take a few moments. Additionally, multiple tags could be written in one session to account for different security practices. For example, if a captain was ok with photographs while sailors were outside the skin of the ship, a tag could be set up to lockdown all of the above features with the exception of the camera.

2. The Sailor Interactions

Once a stepping across the brow of a ship and after showing identification to board, sailors should go to a designated spot near the quarterdeck to scan their phone. They will place their Android device in close proximity to the lockdown tag. An audible beep will be produced by the device, and the application will start. Once the application starts it uses the data provided by the NFC tag to quickly disable features as required by ship's policy. The sailors, when looking at the application, will see which features are disabled in the bottom left corner of the app. When requested at random to show their device, leadership can quickly verify that the phone is in fact locked down in accordance with the ship's captain's policy.

At the end of the day, when getting ready to leave the ship, the sailor will approach the quarterdeck and scan the deactivation NFC tag. This tag will tell the app to re-enable those features that were locked down. The sailors can now receive data, make phone calls, take photos, and do all other activities that were possible before the device was locked down. The sailors will show identification and depart the ship with their phone fully functional and not having ran the risk of photographing or recording classified information.

E. CONSIDERATIONS CONCLUSION

Securing a mobile device requires careful interpretation of DOD policy by a ship's captain, implementation of a policy at the unit level, and a method to manage that policy. The collect or prohibit mobile devices is not always practical and can still lead to security violations. Even with an application that locks down the devices, it is possible that a user might not scan an NFC tag. It is for this reason that ship's force must take it upon themselves to enforce the standard of using the application and spot checking sailors. With this application installed, the possibility for security violations does not go away. However, when a commanding officer has to self report that a sailor sat in combat all day with a mobile device he can at least note that the device was fully locked down and in a state that limited or prohibited the recording of classified data. This is a big step forward in how we manage personal mobile devices, and in future work it will be shown how this should set the stage for implementing methods to take advantage of these powerful tools to assist sailors working on a ship.

THIS PAGE INTENTIONALLY LEFT BLANK

IV. APPLICATION DESIGN, DEVELOPMENT AND TESTING

A. DESIGN CONCEPTUALIZATION

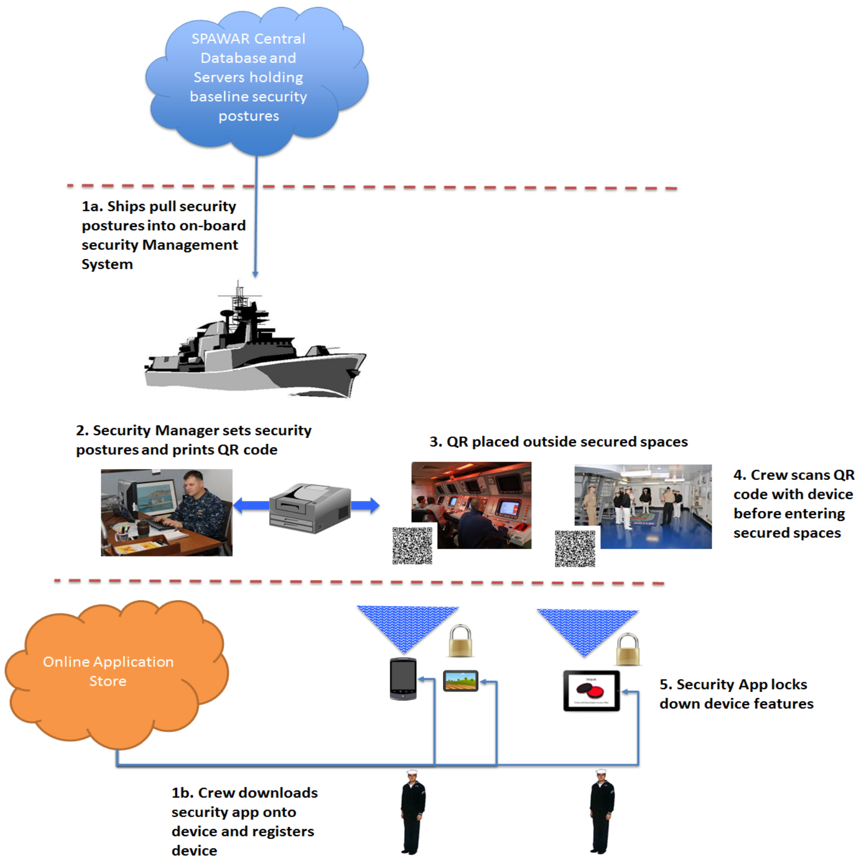
Initial design of our application comes from the notion that there has to be a way to programmatically lock down a mobile device via a mobile application. Specifically, the device should not be able to access features deemed a threat to security while the application holds the state of the device in lockdown. Those features that we find threatening to security based on our experience as security managers include camera, Wi-Fi, Bluetooth, mobile data, microphone and keyboard. We are primarily concerned with mobile device users and any images, recordings, connections, and documentations that they may have initiated or accessed while in a secure space, during a critical exercise, or when otherwise deemed inappropriate by the command.

From this programmatically naïve position, and with literally no computer science background beyond that gained in the courses studied at NPS, our initial solutions originate from our exposure to technology that we use in our daily lives such as our mobile phones and the applications on them. That said, we initially envisioned an application that would be activated and deactivated by scanning a QR code with the device camera. We used this notion to identify requirements and the desired application flow for a project in our Human Computer Interaction that will be discussed further below. Following initial requirements and design consideration study, we commence detailed study in Java and Android programming, leading to several pivot points in our research. Principal among these driving factors is the shift of focus from activation and deactivation via QR code to via NFC. Other discoveries and design changes include the design and layout of Android applications via Google Design guidance, and the policy, class, and methods functionality within Android Studio.

1. Initial Requirements and Design

As part of a course requirement's for Human Computer Interaction (HCI) course, we looked at our security application from a requirements standpoint, developed notions for human computer interactions at each one the identified requirements, refined these notions, and created a sample application implementation without any lockout functionality.[244] We developed the system model seen in Figure 22, whereby the security application controls the state of lockdown on the device, which thereby restricts access to specific features. The model provides for system management via a security manager and overall program management via SPAWAR.

Figure 22. Initial system model for security application


SPAWAR Central Database and Servers holding baseline security postures
1a. Ships pull security postures into on-board security Management System
2. Security Manager sets security postures and prints QR code
3. QR placed outside secured spaces
4. Crew scans QR code with device before entering secured spaces
Online Application Store
5. Security App locks down device features
1b. Crew downloads security app onto device and registers device

Source: Liam J. Dorney et al., "CS3004 Project 1: Mobile Security at Sea (class project, Naval Postgraduate, School, 2015).

a. *Requirements for a Mobile Security Application*

Perhaps one of the most useful set of requirements that evolved out of our research is the expansion of those features that could be deemed a threat to security. We

[244] Liam J. Dorney et al., "CS3004 Project 1: Mobile Security at Sea" (class paper, CS3004, Naval Postgraduate School, 2015).

added the experience of another security manager, LT Sellers, and three foreign nationals (LT Jerel Yam, LT Elmas, and 1st LT Yilmaz) with experience in security. After significant brainstorming and trying to identify every possible threat to security from a mobile device, we identified the following features that would in some way have to be addressed by our security application, either through lockout or banner acknowledgements and agreements:[245]

- Camera—image capture
- Video—video capture
- Microphone—voice recording
- SMS—sharing information
- Document editing—note taking
- Social media—that is accessible by the mobile devices
- GPS—location sharing
- Wi-Fi—Internet access and electromagnetic transmission/radiation
- Bluetooth/infrared—file transmission
- All other forms of wireless data access

This list serves as our primary target list in later development, and interestingly captures the entirety of the requirements of NIST and STIG with respect to connectivity as discussed in Chapter II.

Following requirements identification for features desired to be locked down, we took one step back and attempted to identify system function requirements. Again, relying on our experience with security and mobile devices in the secure environments we had operated in, we attempted to identify as many system functions of our mobile application to enable rapid application deployment and low cost system implementation. From this analysis we identify seven main functions discussed below.

(1) Low Cost Implementation

Beyond development and testing, the security manager's software system will not result in an increase in cost to the Navy because it uses existing hardware already present

[245] Ibid.

on naval vessels.[246] We try to implement a solution that would allow for minimal costs to the Navy. While initially considering QR code implementation, we envision a security manager easily and readily printing out new QR codes and posting them as necessary throughout the ship as security required. While this intent initially holds merit, there are costs associated with development and testing of a secure QR code library, or costs of purchasing and testing of a secure QR code library. The recurring cost would be maintenance of the library and its integration with the application's functions.

The costs of purchasing NFC writers and tags to get a program started are be higher, because a reader/writer would have to be sent to each unit. These devices range in price from around $50 up to $100. Additionally, bulk tags start at approximately 30 cents per tag. Resupplying tags will be a recurring an expense, but the security provided through an NFC implementation overrides the lower cost of simply printing a QR code. This cost, while being higher, seems to be a smart investment and is still not substantial given the cost per tag. It is possible to get a unit set up with NFC hardware and 100 tags for under $100. Additionally, some situations allow for NFC tags to be reused, reducing the number of thrown away tags. The cost per unit by providing a QR code library could presumably be lower, but there are no available encryption or security features for such a library.

(2) Two Factor Device Login for Security

The CO and security manager's interface are protected with two-factor authentication for added security.[247] This design element is generally agreed to by the team and remains a future design feature for our application.

(3) Support for Android and IOS Based Personal Devices

The security system has a mobile device portion that will support Android and IOS based personal devices.[248] This was one of the key components in early system

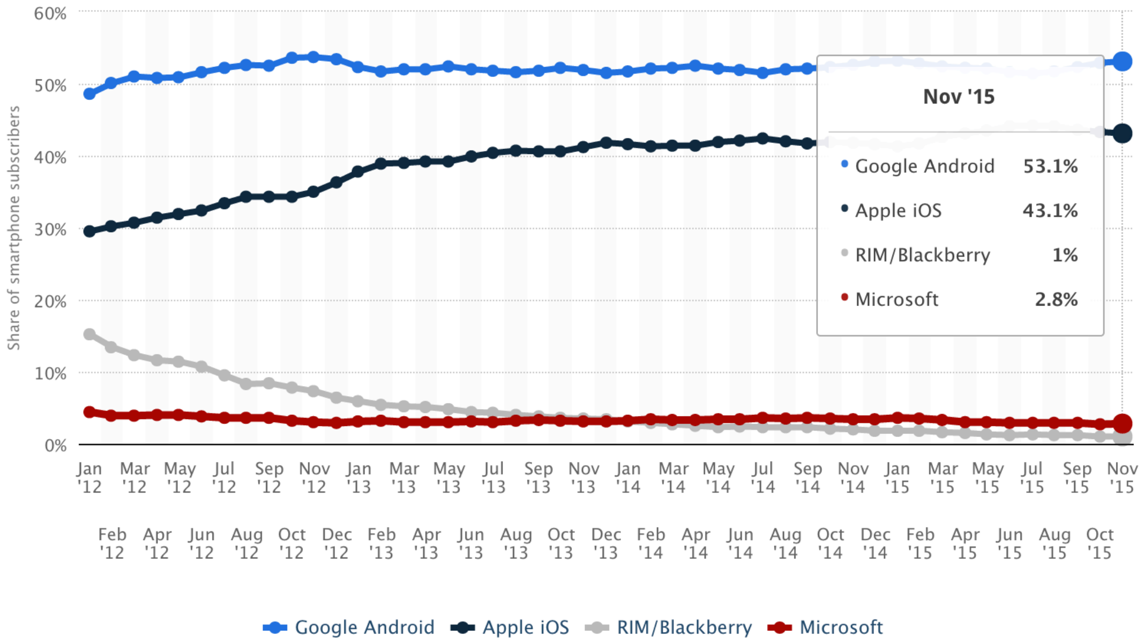
[246] Ibid.

[247] Ibid.

[248] Ibid.

function requirements identification. As seen in Figure 23, by creating an application that covers Android and iOS, the effective coverage for mobile devices in U.S. market would be 96.2 percent.[249] As a cross section of society, we believe that these two OS's equally represent the vast amount of sailor's devices currently and in future use in the Navy.

Figure 23. Smartphone OS market share November 2015



Source: Statista, "Smartphone OS Market Share in the U.S. 2012–2015," Statistic, accessed February 2, 2016, http://www.statista.com/statistics/266572/market-share-held-by-smartphone-platforms-in-the-united-states/.

(4) Offline and Online Functionality

To cater for the ship's network conditions, the security system must take the availability of the wireless network into consideration.[250] This is another key feature identified early that is part of our approach to securing a mobile device. The most important aspect of this feature is the notion that lock and unlock should be possible without prompting via network activity, namely via NFC. On the other side of that

[249] Statista, "Smartphone OS Market Share in the US 2012–2015 Statistic," Statista, accessed February 2, 2016, http://www.statista.com/statistics/266572/market-share-held-by-smartphone-platforms-in-the-united-states/.

[250] Dorney et al., "CS3004 Project 1: Mobile Security at Sea."

feature, is the functionality that comes with connectivity such as the ability to push out updates wirelessly, generate security posture changes remotely, and pulse devices to determine network participation and identification all considerations for future development and research.

(5) Rapid Registration Process for Personal Devices

Registration of personal devices for the ship's crew can be time consuming if there is a considerable number of devices to register, and if the flow of registration is not fluid. The first concept of the system makes use of QR codes for device registration which greatly speeds up the registration process.[251] The initial implementation seeks to have QR codes push registration information to a central database assigning names to device data such as international mobile equipment identifier (IMEI) and the international mobile subscriber identifier (IMSI). While the principle driver of this data push changes in our move from QR code to NFC, the desire to develop a database that stores, secures and is managed by qualified personnel could be an ideal way to track who is registered to use what devices at the command, in what capacity, and whether they are up to date and operating according to the NIST, DISA and STIG security requirements identified and chapter 2. Future work discusses these features more fully.

(6) Central management and Dissemination for Security Postures

The baseline security posture of all the vessels in the fleet can be centrally managed by SPAWAR and central management allows SPAWAR to maintain standards and uniformity for security implementations when necessary.[252] This remains a design goal for our application, as uniformity and central management are key to successful implementation fleet wide.

(7) Allowances for Tweaking of Security Postures on a Ship-by-ship Basis

The COs have the option to tweak the security postures for their ships. This could be useful necessary as every vessel has their own security considerations based on

[251] Ibid.

[252] Ibid.

mission and crew profile.[253] As discussed in Chapter I, the ability of a commanding officer to control the security settings at his or her command is vital to the notion of command in the Navy.

b. Variable Security Postures

Another key finding in the HCI research we conducted was the principle of variable security stances and evaluating how those stances might affect the accessibility of features on a mobile device. As stated above we rely on the security related experience of the members of the team to try to capture the widest range of postures we could envision. We initially identify eight potential postures listed and discussed below:

(1) Low

The application will allow full access to all features on the mobile device, while still allowing it to be tracked on the network.[254] This is the stance (later referred to as a state) of a device when registered, but fully unlocked. This state is also where command specific tailoring could take place, one example being a case where abuse of social media has taken place and a commanding officer may wish to lock out access to social media applications. Said commander could direct the security manager to lock out access to those specific social media applications during working hours, all managed through our application. This is another area for future development with our application.

(2) Home Port Access

While in homeport, the application will allow access to all mobile features outside the skin of the ship. Once internal (and after scanning a QR/NFC code), the application will lock out voice recording, document editing, video and image capture.[255] This state is also one in which further tailoring could take place allowing network access on the mess decks, wardroom, berths or staterooms.

[253] Ibid.

[254] Ibid.

[255] Ibid.

(3) High

The application will fully lockdown all features on all mobile devices. This will be utilized in times of heightened security such as during wartime or highly sensitive operations.[256] Furthermore this state represents the most restrictive stance of the application applying controls over the device.

(4) Visitor

While in homeport and overseas, tours of U.S. naval warships occur frequently. This setting is for all visitors who are visiting the ship at port. It lockouts all features of the device, and provide near real time access attempt notifications to the security team. It also requires registration and application download prior to commencement of any tour.[257] A significant amount of discussion went into topic, as the potential for release of our application as open source presents certain security threats, such as manipulation and introduction of malware of spyware. One consensus calls for the design of a visitor application that stood alone from the other DOD based application. This of course presents its own challenges as legalities with manipulating the settings of an individual's phone could raise alarms with the individual or their organization. Any common sense individual can also easily see how this could be an issue for a high-ranking dignitary visiting a ship on short notice in a foreign port.

(5) Secret and Higher

Full lockdown of device prior to accessing the space.[258] This and the High setting discussed above were chosen in the design phase as the starting point for development with the rationale that if we could fully lock down the device we should be able to tailor those lockdown executions based on the settings listed herein. In this thesis we apply what we agree to as the most threating features to security as the starting point, namely camera, Wi-Fi, Bluetooth, microphone, and mobile data.

[256] Ibid.

[257] Ibid.

[258] Ibid.

(6) Video and Image Sensitive

For evolutions deemed appropriate by the commanding officer and his team such as missile launches, classified ships maneuvers, video and image capture features should be locked down.[259] Every member of the team identified a time at their command that the leadership had expressed concern over the open availability of mobile device image and video capture. One such event was the launch of a tomahawk missile in 7th Fleet. While thoroughly documented and discussed on the Internet, the full video capture of all phases of launch was captured by crew members spectating and resulted in significant delays in releasing the phone back to crewmembers due to security evaluations of the video data. This is one very public example, but any individual with military experience can envision times when video and image recording could threaten the security of the exercise, sailors, of the command.

(7) Underway

Lockdown camera, GPS, and wireless access as deemed necessary by the commanding officer and his security team.[260] This is a specific posture identified by the American members of the team as essential given our experience with sailors accessing their mobile devices during high risk underway operations such as sea and anchor details, underway replenishments, or flight operations. In all the situations listed, individuals have been found utilizing their devices when in proximity of cell or data signals, not only distracting from their duties, but in also in the case of GPS providing potential signaling data to adversaries.

(8) Silence

All the devices are set to be silent, by disabling their speakers and headset outputs.[261] This is a setting identified by the entirety of the team and speaks to the need for device silence in sensitive meetings where high-ranking individuals are speaking or

[259] Ibid.

[260] Ibid.

[261] Ibid.

being briefed. Everyone spoke to separate instances where a silence option would have provided critical backup to a lower ranking individual whose phone alarmed or rang while someone several pay grades above them was speaking or being honored.

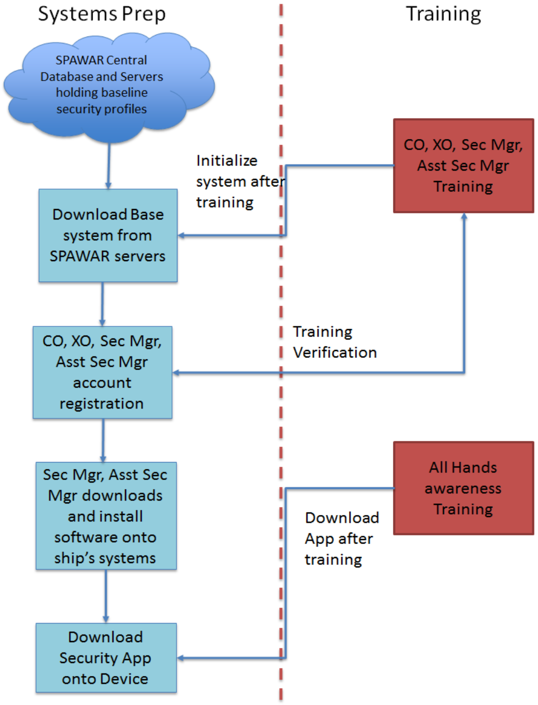
c. Application Employment

As part of larger adoption of our application, we attempt to identify as many critical stages as possible, where the program and associated application require leadership training, interaction and use. As with all things military, we identify early that the need for training on any new system or process could not be avoided. We therefore try to keep our application and its associated process as basic as possible, and have operation and flow occur as closely as possible to applications and systems with which most would have experience. As a result, we identify five specific stages and identified the critical steps in each stage to ensure the intended goal for the user is met, as discussed below.

(1) System Initialization

With all of the members of our team having experienced the implementation of new software or programs at previous commands, we all felt comfortable with targeting initialization as a critical stage in the utilization of the security application. We therefore brake initialization up into two parts, System preparation and Training, making the former dependent on the latter so that opportunities for training on the system cannot be bypassed by leadership, security personnel, or users as seen in Figure 24.

Figure 24. System initialization

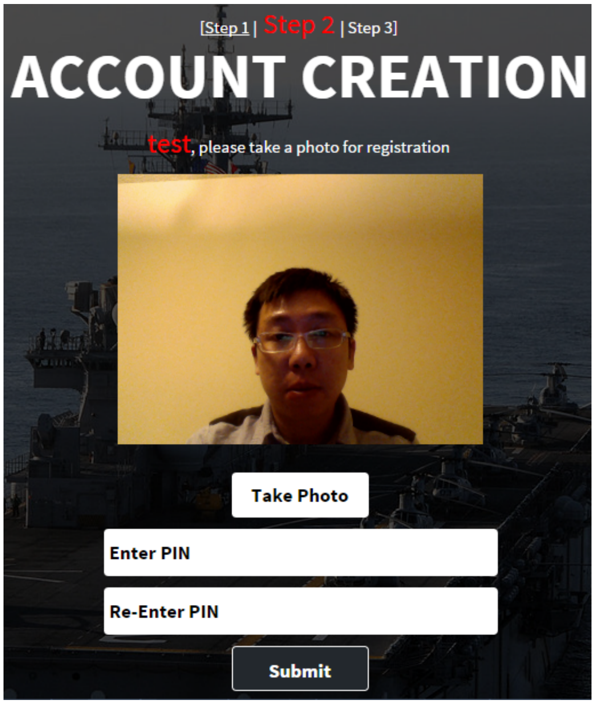

Systems Prep
Training
SPAWAR Central Database and Servers holding baseline security profiles
Download Base system from SPAWAR servers
Initialize system after training
CO, XO, Sec Mgr, Asst Sec Mgr Training
CO, XO, Sec Mgr, Asst Sec Mgr account registration
Training Verification
Sec Mgr, Asst Sec Mgr downloads and install software onto ship's systems
All Hands awareness Training
Download App after training
Download Security App onto Device

Source: Dorney et al., "CS3004 Project 1: Mobile Security at Sea."

One of the key events identified in this stage is the need to have a source of baseline data from which to draw the initial system software. This data storage and management is a significant piece to any major technology implementation in the Navy. We propose it for future work as an opportunity for SPAWAR or 10th Fleet to manage and direct software implementations and updates to our application.

The other key piece to initialization is the notion of two types of registration, the first for commanders and security managers, and the other for the users or ships force. We envision the registration in the case of leadership and security personnel involving facial recognition and user password combinations as seen in Figure 25.

Figure 25. Commanding officer and security manager facial recognition and username password screen



[Step 1 | Step 2 | Step 3]

ACCOUNT CREATION

test, please take a photo for registration

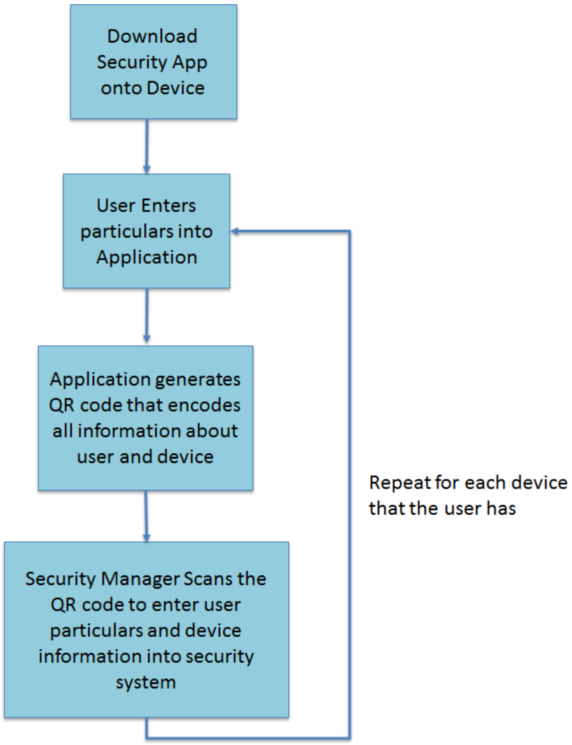
Source: Dorney et al., "CS3004 Project 3: Mobile Security at Sea."

This provided for two-layer security access to setting for said individuals, but again requires database storage and management for name, facial data, password, and username information.

(2) All Hands Device Registration

In this critical stage, we identify the need to identify and register each device into the command database. The user inputs their username, a password, and then the system generates a QR code that is scanned by command security management for input into the security system. Our goal in this stage is to ensure that the database accurately captures all of the user's information and correlates it to device specific information. We place a simple repeating step to the process so that all the devices a user owns are registered and reported to the management suite of software as seen in Figure 26. In this process, each device registered generates a new QR code that is individually uploaded to the command managed database for storage and tracking as required. Registration via QR code generation presents database management and security related challenges. For that reason, we now envision the application reporting features of the device including user name and division on initial connection to the network following application installation on a device. By these means, the application sends a report to the system containing IMEI, IMSI, User ID etc., and is stored within the system suite for access in report generation and device monitoring.

Figure 26. Device registration process

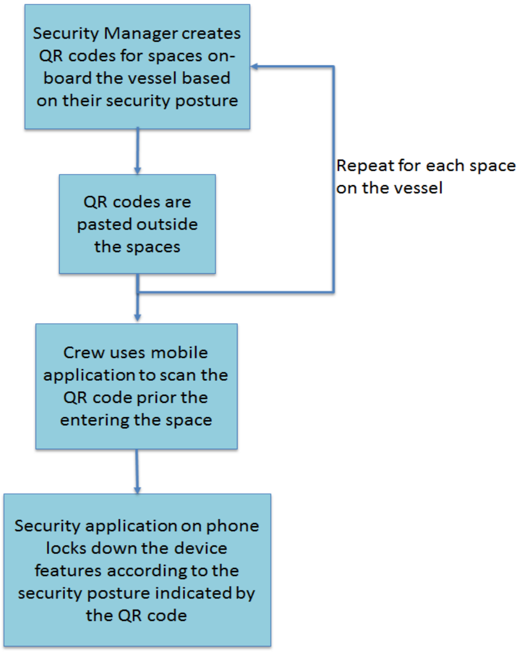

Download Security App onto Device
User Enters particulars into Application
Application generates QR code that encodes all information about user and device
Security Manager Scans the QR code to enter user particulars and device information into security system
Repeat for each device that the user has

Source: Dorney et al., "CS3004 Project 3: Mobile Security at Sea."

(3) Space Specific Lockdown

This feature spans initial QR code and eventual NFC tag implementations of our application design. The ability of a security manager and their respective command to determine specific postures and have those postures translate into device security settings, then rapidly deploy that setting mechanism to the space in question is deemed essential to the success of our application by all parties involved in our research. Initially we envision a QR code posted outside a space that could be readily updated by the security manger as seen in Figure 27.

Figure 27. QR code implementation of security settings at various spaces in a command


Security Manager creates QR codes for spaces on-board the vessel based on their security posture
QR codes are pasted outside the spaces
Repeat for each space on the vessel
Crew uses mobile application to scan the QR code prior the entering the space
Security application on phone locks down the device features according to the security posture indicated by the QR code

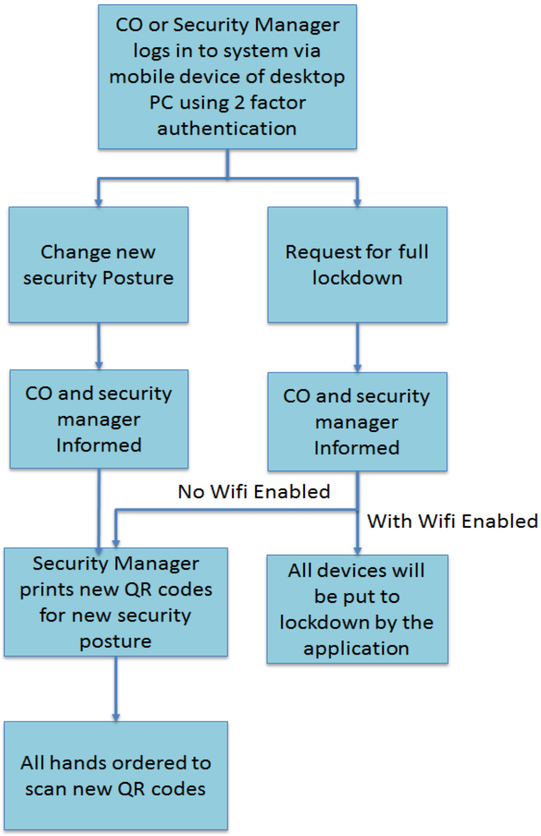
Source: Dorney et al., "CS3004 Project 3: Mobile Security at Sea."

This seems ideal, but based on the size and management requirements of useable QR generation software, combined with the storage and security of QR code libraries, we migrate to utilization of NFC writer and tags. In place of generating a QR code, posting outside a space, and having users scan it, we write to an NFC tag that initiates the security features built into our application and tailored by the security manager. These tags are inexpensive, disposable, and are able to be written so that they are read only, preventing any future access to data on the tag. Writing is also a non-issue, as multiple downloadable mobile application exits for tag writing, as does hardware devices that can be connected to a management suite. Furthermore, tag writing can be limited to specific users, allowing the security manager to re-write a tag as needed without fear of compromise from other individuals with malicious intent.

(4) Real-Time Security Posture Change

Having been in multiple scenarios whereby the commanding officer or his designated official have to make a command wide announcement to secure mobile devices, the need for a real time push to change the security setting on every registered uses device is deemed vital. To this end, we identify a Wi-Fi enabled and non-Wi-Fi enabled scenario for rapid changes to security setting on devices as seen in Figure 28.

Figure 28. Real-time security posture change


CO or Security Manager logs in to system via mobile device of desktop PC using 2 factor authentication
Change new security Posture
Request for full lockdown
CO and security manager Informed
CO and security manager Informed
No Wifi Enabled
With Wifi Enabled
Security Manager prints new QR codes for new security posture
All devices will be put to lockdown by the application
All hands ordered to scan new QR codes

Source: Dorney et al., “CS3004 Project 3: Mobile Security at Sea.”

Again, this early design incorporates the use of QR codes. In a Wi-Fi disabled environment, this involves printing out new QR codes and posting them throughout the command, or carrying pre-printed QR codes. For the pivot into NFC, we envision a predetermined setting written to a tag that is placed in a position of easy access such as a manager’s clipboard or on the bridge log. These settings could be any of the above postures or a tailored version of any of them and written to a tag. Having several tags labelled and adhered to the back of a clipboard or log is one fast and effective method to rapidly change postures in a situation where Wi-Fi is not enabled. In a case where Wi-Fi is enabled, a system wide push notification updates all connected devices settings as desired by the commanding officer or security manager. This is not currently built into our application but is a significant consideration for future work as discussed in Chapter V.

(5) Report Generation

Having all spent time answering to senior officers, most military personnel understand how often the manager requires a system report or wants to know the status of a system or process. For that eventuality we incorporate report generation into the critical stages of our application. This feature is not part of the current application implementation, but identified as a significant area for future development in our application. We envision the ability to press a button interface within the application and have the system report back all desired information, such as number of users, users by setting and posture, and users and associated devices. This again points to the need for database management and is another area for future work.

B. EARLY DEVELOPMENT

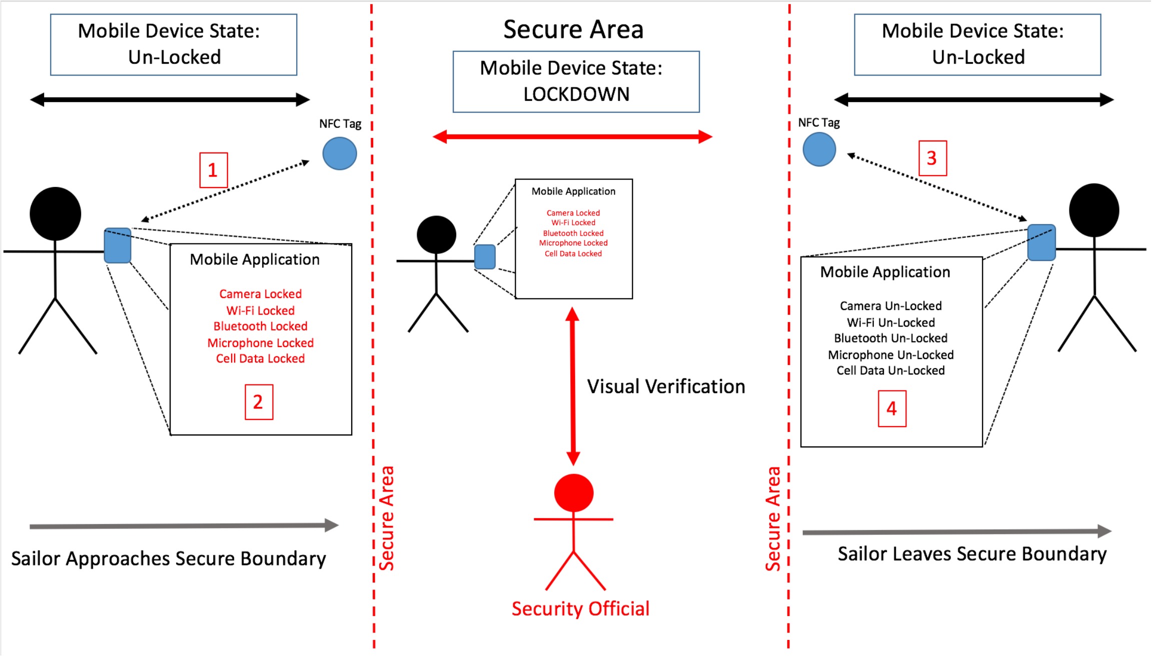
At onset we had no background in Java, the underlying programming language used in Android programming, or the Android environment itself. This proved to be a key setback as our early attempts to manipulate the settings and features we identified in conceptualization were not successful. As discussed in Chapter III, we commence our development in a major technology shift period for Android development. At the time of initial conceptualization, Eclipse was still in widespread use and for the most part no local resources or instructions were available for the industry shift to Android Studio. Several hours were spent in the research phase of identifying each feature we sought to lockdown, accessing the classes and methods that allowed us to manipulate those features, only to find that through the Eclipse IDE, numerous features were no longer applicable or worked across any reasonable range of devices, either emulated or actual. In fact, after two months of attempting to program Android in Eclipse, we were forced to transition to Android Studio in order to access the full power of Androids OS.

Once inside Android Studio our principle approach to tackling each feature involves creating an instance of each feature. For example, we create an instance of the camera, display it on the screen, then shut it off. This is done for camera and microphone but proves impossible in the case of Wi-Fi, Bluetooth, or cell data. In these three cases, we have to implement a toggle switch that enables on and off functionality, as we cannot initially programmatically toggle each feature. Again this leads us to a shift in our focus, as we are convinced there has to be a method to restrict access to these features beyond building in toggle switches. As a note the encapsulation method we built in our initial attempts at application design are not conducive to initiation via QR code or NFC. Specifically, as we progress through Wi-Fi and Bluetooth toggling, we are encapsulating the event toggle inside other toggle events, activated by buttons, leading to convolution of event logic. This logic flaw progresses into a situation where button presses are not connected to all the events for the desired outcome of the event, for example "lockdown" results in turning Wi-Fi on and Bluetooth off. This again drives us to shift our focus to build the interaction event, i.e., the swiping of an NFC device, and then encapsulate the features we wanted to access inside that event. This provides a critical point for development moving forward. We now realize that we have two very distinct events to manage and code for: (1) activity driven system response such as NFC swipe, and (2) lockdown and unlock. Furthermore, we need to focus on seamless integration of the activities that drove system responses, avoiding logic flaws and outdated class features.

C. ADVANCED DEVELOPMENT

Realizing that we cannot possibly capture all the features and critical stages in our application without establishing the basic functionality of lockdown and unlock controlled by NFC device, we have to scale down the scope and focus on the lock/unlock activity first. In doing so we develop a use case scenario to better understand the steps we want to accomplish in the act of lockout out the mobile device. We then create sequence diagrams for those steps, which further assist us breaking down how we go after each part of the application. Once we define a scaled down version of our goal, visualize how this goal might be achieved, and what steps are required, we move into advanced development. Through early analysis of requirements, critical stages, and visual interpretation of steps, we are able to step into development at a streamlined point of origin, where our application is not trying to solve the entire problem set by meeting all the requirements in one place, rather addressing a very specific set of lockdown features requiring simple UI. In Figure 29, we break down the four basic steps we wanted to accomplish with our application as a user entered and then exited a secure facility. Note that at any point with the application running in the foreground, the status of the features controlled by the application is visible to anyone who wished to verify the status of the device.

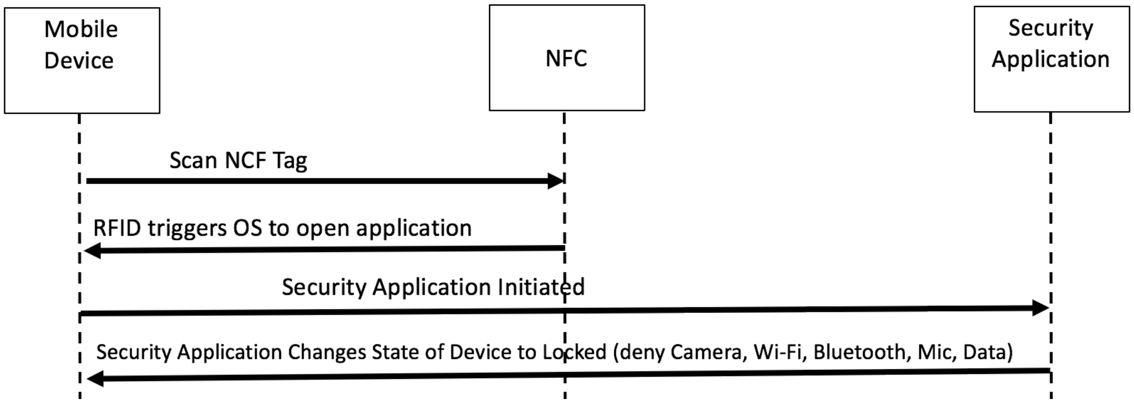
Figure 29. 4 step use case diagram for use of security application



In this use case diagram, a user has the security application on their device and progresses through a secure area. Locking and unlocking the device is accomplished in four steps: (1) swipe NFC tag, (2) Application locks features on device, User enters the secure space with device locked down, (3) user exits the space then swipes an NFC tag, (4) application unlocks access to device features. Note the five features we are trying to manipulate in this implementation are Camera, Wi-Fi, Bluetooth, Microphone, and Cell Data.

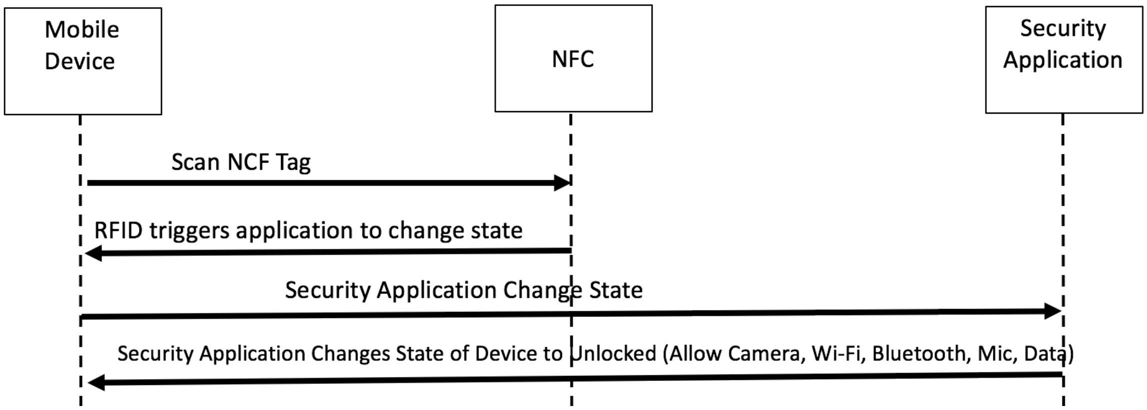
The desired interactions in each step are captured in sequence diagrams shown in Figures 30 (lock) and Figure 31 (unlock). As stated above, breaking down the interactions as we understand them and identifying the interaction we are able to solve critical issues related to how we are attempting to manipulate the mobile device state. For instance, the NFC itself triggers the application through the OS, not triggering individual specific features, which has been the premise of earlier research and development efforts.

Figure 30. Lockdown interaction diagram


Mobile Device
NFC
Security Application
Scan NCF Tag
RFID triggers OS to open application
Security Application Initiated
Security Application Changes State of Device to Locked (deny Camera, Wi-Fi, Bluetooth, Mic, Data)

A simplified interaction diagram that captures the four step interaction between the mobile device, the NFC tag, and the security application.

Figure 31. Unlock interaction diagram


Mobile Device
NFC
Security Application
Scan NCF Tag
RFID triggers application to change state
Security Application Change State
Security Application Changes State of Device to Unlocked (Allow Camera, Wi-Fi, Bluetooth, Mic, Data)

The reversal of the lockdown noted in Figure 29, this interaction diagram shows the unlock interactions between the mobile device, NFC tag, and the security application.

As we step into the locking and unlocking of features we realize that the Android programming environment contains literally thousands of super classes, classes, and methods that are used throughout the activity life cycle to achieve the ends of each activity. In our early more basic research we assumed that we would be able to utilize a feature, such as opening an instance of the camera, and then enveloping that instance in a lockdown activity. Interestingly enough we have discovered that we need to access very specific classes that further access the lockdown function associated with a feature. Even more challenging is the fact that certain features are not directly lockable, that is to say the OS cannot directly deny access to these features, based on Android principle of least privilege. A reminder from Chapter III, according to this architecture, no one application can utilize any one resource at a higher priority than another.[262] Furthermore, this OS rule also restricts access to classes and methods according to permissions. These two aspects are discussed more thoroughly below.

1. Lock and Unlock Access

After numerous attempts to simply lock out access to features, and investigating Androids principle of least principle we identify one crucial piece of development information: various system permissions are required to execute these lockdowns. The next critical step is ensuring that the locking methods we use fully captured each feature. In many instances, we catch one feature during startup, but do not address it if the feature is already running, or while the device is changing state. These are flushed out through our continuous efforts to capture every event related to each feature, and they are discussed below under intents. With respect to actually locking out a feature however, permissions themselves define the scope of method usage and ultimate device feature access.

a. Permissions

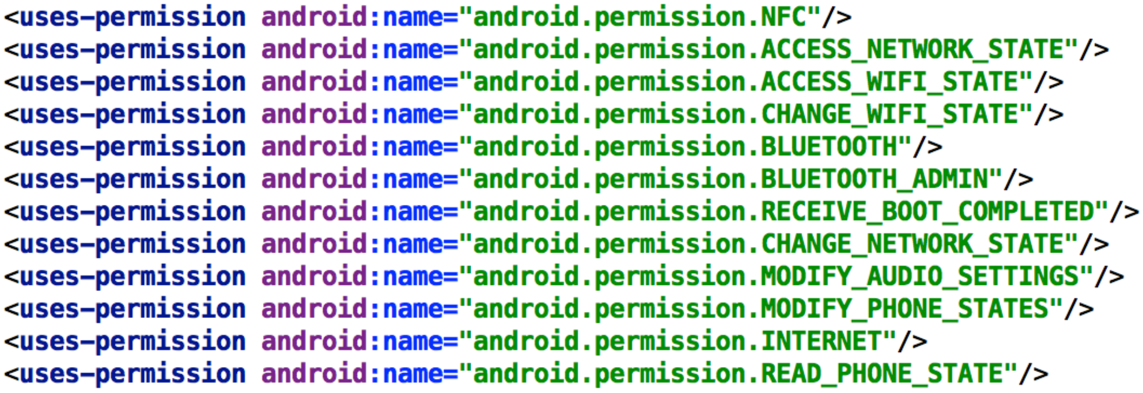
Permissions are coded into the AndoridManifest.xml, which is the root file for the application.[263] This file contains among other things the activities, broadcast receivers, services, and links those such items to an associated permission. In a more specific sense, and a direct finding in our research, the most thoroughly written android activities and broadcast receivers will do exactly nothing if they do not have the proper permissions, and low level debugging is required to catch situations where either of those aspects of a program (activities or broadcast receivers) do not have the proper permissions. The second and perhaps more important factor with respect to permissions is various groups of permissions. As stated by Google, the two most common groups are normal

[262] Developer, "Application Fundamentals."

[263] Developer, "App Manifest," accessed January 18, 2016, http://developer.android.com/guide/topics/manifest/manifest-intro.html.

permissions and dangerous permissions.[264] Normal permissions include any permission that allows access to any data that is native to the sandbox of the application, such as setting an alarm, or launching a web browser.[265] Dangerous permissions on the other hand are those permissions that an application seeks such that user data is affected, privacy is threatened, or system wide effects are expected as a result of the permission given. This presents a very critical turning point in our research as our application initially seeks to use a dangerous permission: we want to lock out access to features system wide and prevent the user from manipulating that setting. As a result, we re-work our attempts at accessing those permissions by using others as seen in Figure 32 and as discussed below.

Figure 32. Security application list of permissions



```
<uses-permission android:name="android.permission.NFC"/>
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE"/>
<uses-permission android:name="android.permission.ACCESS_WIFI_STATE"/>
<uses-permission android:name="android.permission.CHANGE_WIFI_STATE"/>
<uses-permission android:name="android.permission.BLUETOOTH"/>
<uses-permission android:name="android.permission.BLUETOOTH_ADMIN"/>
<uses-permission android:name="android.permission.RECEIVE_BOOT_COMPLETED"/>
<uses-permission android:name="android.permission.CHANGE_NETWORK_STATE"/>
<uses-permission android:name="android.permission.MODIFY_AUDIO_SETTINGS"/>
<uses-permission android:name="android.permission.MODIFY_PHONE_STATES"/>
<uses-permission android:name="android.permission.INTERNET"/>
<uses-permission android:name="android.permission.READ_PHONE_STATE"/>
```

Android Studio permissions listed in the AndroidManifest.xml file showing the permissions utilized in our security application.

(1) Dangerous Permissions

Note that writing permissions that Google and the Android IDE identify as dangerous requires not only permission written in the AndroidManifest.xml but also an intent that sends an approval request to the user anytime that permission is accessed.[266]

[264] Developer, “System Permissions,” accessed February 3, 2016, http://developer.android.com/guide/topics/security/permissions.html.

[265] Ibid.

[266] Ibid.

This is a feature that comes as part of the Android update in API 23. Prior to this update, authorizations were granted at installation of an application. Google and numerous forums identify the installation acknowledgements for dangerous permission granting as a security flaw that takes control of the device away from the user.[267] This is an important finding as it changes the nature of our application. If every time an attempt to change system wide setting results in user prompting for each individual feature to grant authorization for setting change, our application becomes burdensome, inefficient, and something people generally try to avoid using. Note that normal permissions (non-dangerous) are still handled at installation and do not usually require user authorization, making them a more appropriate target for use. Given this challenge, we pivot late in development and take a different approach on just about every feature we are attempting to control via dangerous permissions, resulting in the need for device policy administration and broadcast receivers, discussed in the section under that name.

(2) Normal Permissions

Our application, in the current iteration, uses only normal permissions to avoid the added interactions that are required when using dangerous permissions as seen in Figure 32. There are a couple of lesser used permissions we employ that are worth identifying here, as until we discovered their need and associated potential, our application was functioning in a diminished capacity. First among them is RECEIVE_BOOT_COMPLETED. This permission enables our application to receive system wide notification that the device has completed booting, essential for tracking of a restart of the device and our attempts at maintaining a listening state within device for attempts to reboot out of the lockdown/unlock state the application was in prior to shutdown.

Another permission worth noting is MODIFY_PHONE_STATES. This permission is required for a few of our features, but is considered a diminished permission with reduced capacity. As discussed above, within the principle of least privilege this permission is only able to touch the non-dangerous states of the phone, i.e.,

[267] Ibid.

those that reside in the sandbox of the application. That said this method grants access to monitoring and not actually modifying phone states.

The NFC permission (android.permission.NFC) is worth discussing as well in so far as it is an older (API 9) permission, but a highly effective one. It contains the permissions for full functionality and interaction with NFC tags. Given the scope of available tag types (four unique types), modes (peer-to-peer, read/write, card emulation)[268] and interaction methods, this normal permission is very self-contained in that it grants full access to all NFC capabilities. As an example, compare NFC to Wi-Fi. Our manipulation of Wi-Fi required extensive research to identify effective permissions, as opposed to the all-inclusive design and rapidly employed NFC permission. Also notice in Figure 32 that both NETWORK_STATE and WIFI_STATE are required for Wi-Fi features we access, and further that both require ACCESS and CHANGE components to accomplish the same functions NFC accomplishes in one permission.

2. Methods of Control

Having run into roadblocks with permissions we pivot and seek out other ways to control the five features in our initial implementation. This change of how we think about controlling features is key to our success as it breaks our application into three distinct approaches to achieve lockdown. First, we have discovered the device policy manager that Google developed for an enterprise solution for mobile devices. Using this approach, we realize that specific features can be controlled at the system level within a managed profile that is created and approved at application installation. Second, for the features not built into the Device Policy Manager we have decided to build broadcast receivers to receive notification of intents (such as Wi-Fi state change), and then allow the application to enforce a simple on or off setting related to those intents. And third, we have discovered that one specific feature is controllable programmatically without the need for a managed profile or broadcast receiver, namely the microphone.

[268] Gerald Madlmayr, “NFC Development & Consulting,” NFC Development Consulting, March 8, 2011. http://www.nfc.cc/technology/nfc/.

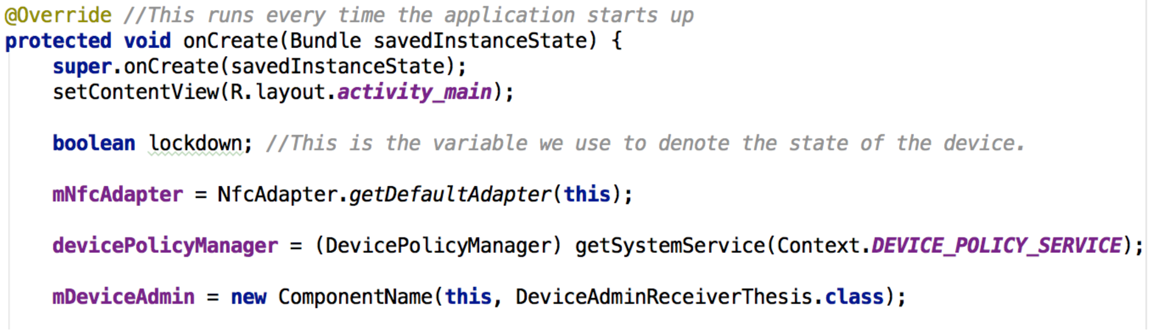
a. Device Policy Manager

The Device Policy Manager class provides the deepest level of control for the accessible features, as the permissions and services controlled therein operate at the root level in the OS. This class is supported through the Device Administration API, whereby the device installs the application as an administration application and the user grants system level access to it.[269] In the case of our application this access is granted in return for being able to utilize a mobile device at work, adhering to the reward for access guidance from Google. In any case, upon discovering this API, we immediately explored the features that we can enforce with its system policies and privileges. Unfortunately, the only feature of use to our application included in the current version of the API is the camera. Several other useful features are available to be programmed into the profile, such as wipe features for number of password attempts and lost device situations, but they are not within the scope of our current research.

Within the Device Policy Manger, locking and unlocking of the camera is very straight forward and dependable, as expected given the system level privileges inherent to its implementation. As seen in Figure 33 by casting the SystemService context of the Device_Policy_Service into DevicePolicyManager, built in class features take care of system prompts to the user for acknowledgement of privileges, and present the administrator authorization screen, buttons, and handlers for a UI. Given the degree of built in features for this policy administration, this is the preferred means of controlling all the features on the device we seek to control. For this reason, and for others noted throughout the broadcast receiver section, we recommend DOD and Navy approach Google to have other feature policy restrictions built into Device Policy Manager as the most effective way to ensure that devices are fully locked down.

[269] Developer, “Device Administration.”

Figure 33. Device policy administrator construction



```
@Override //This runs every time the application starts up
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);

    boolean lockdown; //This is the variable we use to denote the state of the device.

    mNfcAdapter = NfcAdapter.getDefaultAdapter(this);

    devicePolicyManager = (DevicePolicyManager) getSystemService(Context.DEVICE_POLICY_SERVICE);

    mDeviceAdmin = new ComponentName(this, DeviceAdminReceiverThesis.class);
```

Construction of the devicePolicyManager in the OnCreate class.

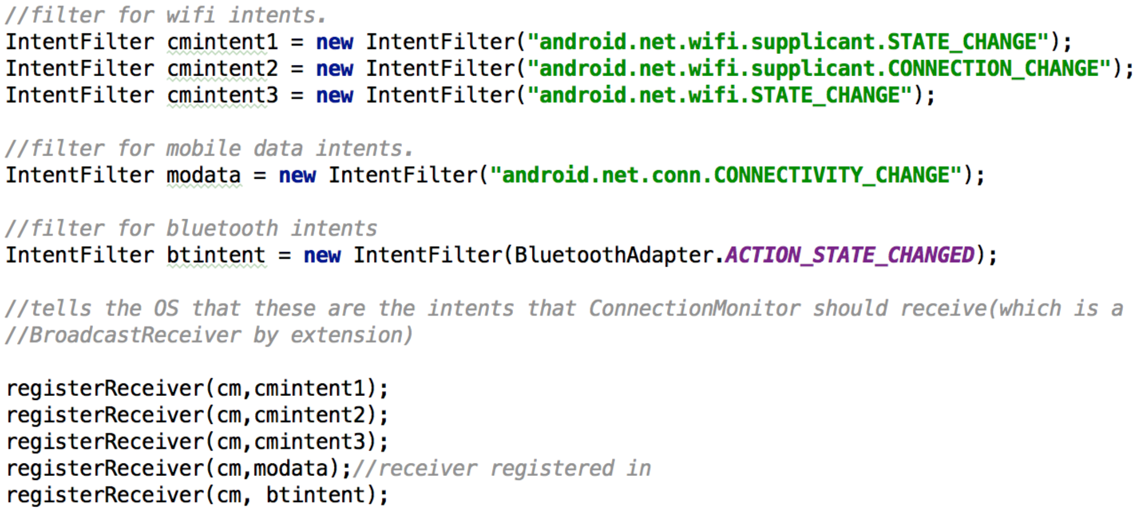
b. Intents and Broadcast Receivers

Given the lack of access to all features we seek to control beside camera (Wi-Fi, Bluetooth, Mobile Data and Microphone) with in the Device Policy Manager class, we had to find another way to have our application be aware of device access attempts, without requiring system level “dangerous” permissions. This drove us to look deeper at the actual components of the Android application architecture, and specifically utilize the Broadcast Receiver component. The amount of time taken to delve into the implementation of a broadcast receiver in our application cannot be overstated, as the complexity and system knowledge required to identify, register, and wire up responses for the activity driven intents that the broadcast receiver must listen for is extensive and multi-facetted. We include specific examples below as we implement and test, implement and test, along the Agile programming method. In layman’s terms, by utilizing the broadcast receiver method of notification we essentially tell the system our application wants to be notified of any attempt to access a feature, like the proverbial hand in the cookie jar causing an audible alarm. We then programmatically take some action based on that notification to either initiate lockdown/unlock, or ensure the desired state is still in place and enforced via simple on and off commands.

(1) Intents

As stated above the most complicated factor in getting our application to function properly outside the initial plan whereby we accessed dangerous permissions is by utilizing broadcast receivers and associated intents. In trying to build broadcast receivers for the application countless hours were spent trying to determine the appropriate, effective, and authorized intents filters for the system broadcasts of feature access attempts (i.e., turn on Wi-Fi). Once the desired intent filter is found and assigned, it must be registered with the appropriate broadcast receiver. Our applications intents are registered in the ConnectionMonitor (extends broadcast monitor) class that contains all the methods that the program will execute when the OS sends it a filtered intent. Notice in Figure 34 that the various features we access require different intents. The most challenging system broadcast to address is Wi-Fi, which has multiple intents available in the WiFiManager class. Our initial attempts to trigger receiver responses to Wi-Fi.STATE_CHANGE were unsuccessful at catching more fine-grained system state changes related to Wi-Fi state. For that reason, we add the designated fine-grained supplicant states that address specific access to the security protocols of Wi-Fi initiation (WPA) via Wi-Fi.supplicant.STATE_CHANGE and .CONNECTION_CHANGE.

Figure 34. Security application intents



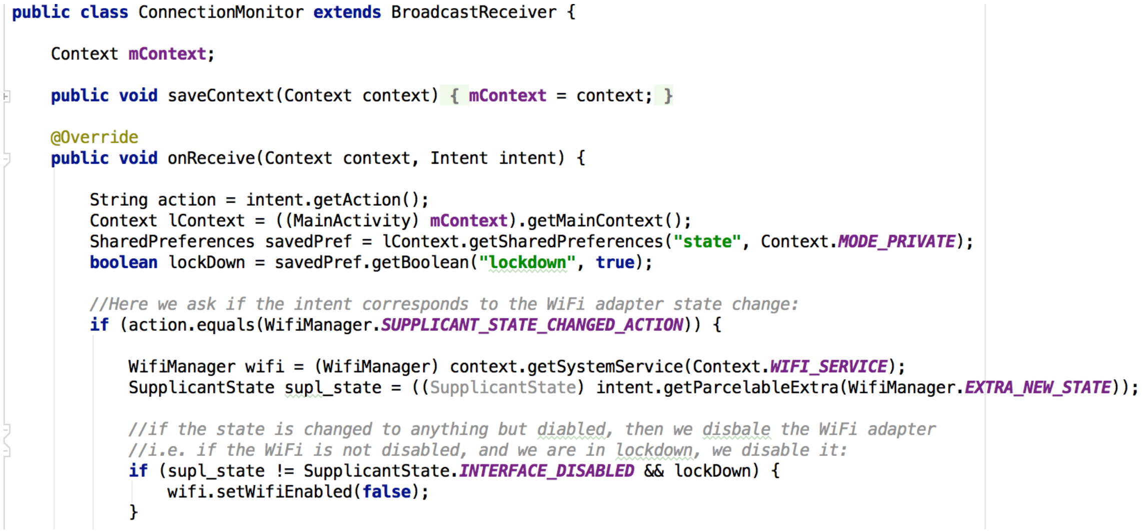
```
//filter for wifi intents.
IntentFilter cmintent1 = new IntentFilter("android.net.wifi.supplicant.STATE_CHANGE");
IntentFilter cmintent2 = new IntentFilter("android.net.wifi.supplicant.CONNECTION_CHANGE");
IntentFilter cmintent3 = new IntentFilter("android.net.wifi.STATE_CHANGE");

//filter for mobile data intents.
IntentFilter modata = new IntentFilter("android.net.conn.CONNECTIVITY_CHANGE");

//filter for bluetooth intents
IntentFilter btintent = new IntentFilter(BluetoothAdapter.ACTION_STATE_CHANGED);

//tells the OS that these are the intents that ConnectionMonitor should receive(which is a
//BroadcastReceiver by extension)

registerReceiver(cm,cmintent1);
registerReceiver(cm,cmintent2);
registerReceiver(cm,cmintent3);
registerReceiver(cm,modata);//receiver registered in
registerReceiver(cm, btintent);
```

MainActivity.java file intents that we declare and then register with the broadcast receiver for OS notification suring system wide broadcasts.

(2) Broadcast Receivers

The aim here is to capture every possible system broadcast and drive some reaction to that broadcast. We take the approach that a system broadcast should trigger our application to verify the state of the feature that triggered the notification, then check the state of our application (either locked down or unlocked), and then finally trigger the placement of the feature accessed into the state it should be in according to the state of our application (if they differ). We essentially listen for an attempt to change Wi-Fi or Bluetooth system state, determine if the state change is in accordance with our application current state (locked or unlocked), and then have our application step in to place the feature in the desired state as necessary. By doing so, we catch any other applications attempts to access the intents associated with turning Wi-Fi or Bluetooth on or off, and have our application intervene and take control of those features according to what state the device was in. In a running example, if a social media application attempts to turn Wi-Fi on, the OS broadcasts that applications intent system wide to all registered intents. That broadcast is picked up by our applications registered intent filters for Wi-Fi. In turn our application activates and resumes (in the background) programmatically setting the state back to the state recorded in the security application's static memory, essentially revoking the Wi-Fi state change attempt out of the social media application's control (see Figure 35).

Figure 35. Connection monitor broadcast receiver



```
public class ConnectionMonitor extends BroadcastReceiver {

    Context mContext;

    public void saveContext(Context context) { mContext = context; }

    @Override
    public void onReceive(Context context, Intent intent) {

        String action = intent.getAction();
        Context lContext = ((MainActivity) mContext).getMainContext();
        SharedPreferences savedPref = lContext.getSharedPreferences("state", Context.MODE_PRIVATE);
        boolean lockDown = savedPref.getBoolean("lockdown", true);

        //Here we ask if the intent corresponds to the WiFi adapter state change:
        if (action.equals(WifiManager.SUPPLICANT_STATE_CHANGED_ACTION)) {

            WifiManager wifi = (WifiManager) context.getSystemService(Context.WIFI_SERVICE);
            SupplicantState supl_state = ((SupplicantState) intent.getParcelableExtra(WifiManager.EXTRA_NEW_STATE));

            //if the state is changed to anything but diabled, then we disbale the WiFi adapter
            //i.e. if the WiFi is not disabled, and we are in lockdown, we disable it:
            if (supl_state != SupplicantState.INTERFACE_DISABLED && lockDown) {
                wifi.setWifiEnabled(false);
            }
```

Source: Yazan, "Prevent Android Phone from Connecting to WiFi Network," Stackoverflow, November 12, 2014, http://stackoverflow.com/questions/26687211/prevent-android-phone-from-connecting-to-wifi-network-unless-my-app-approves-it. Security Application Connection Monitor Class with onReceive method and applicable operations for checking the intent, the state of the Wi-Fi adapter, and putting it in the correct state. Modified from stackoverflow.

Another key to our research is building the broadcast receiver for the NFC interface. In fact, as stated above early development focused on QR code interface with our application. Through working with how a broadcast receiver would respond to a defined QR code our research drove us to look for a more meaningful system trigger. We find that QR codes need to be very specifically formatted, requiring extensive generation software and user experience, especially given that we seek to have our application respond to various different and rapidly changing QR codes. We question the immediate feasibility of the research required for the various aspects of QR code library security, management from a database standpoint, and a user in the fleets ability to easily navigate these issues. These issues combined with the extensive broadcast receiver issues we have with triggering actions from any QR code interaction, never mind specific QR codes, ultimately drive us to use NFC triggered events.

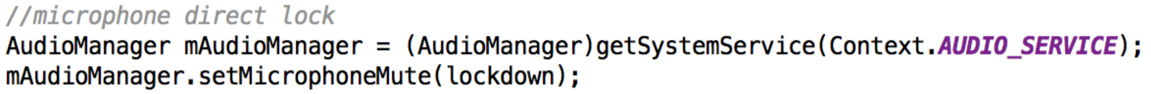
NFC has instantly noticeable advantages from a programming standpoint. First and foremost, the ease with which permissions, intents, and broadcast receivers seamlessly integrated into the OS. An example is the initiation of system responses instantly upon creating and registering intents, and building the most basic of receivers.

This pushes our research in a security related direction as we seek to protect the application from activation via rouge NFC tags. The solution we propose is to write a simple binary string to the NFC tag "Hello, World!." Our receiver looks for any string, and makes the required state check and change actions upon reading it. We note this as an important feature, as the application could be modified in future implementations such that the NFC receiver only reacts to specific strings. This allows for the use of secure hashes to be used in place of a simple string, enabling improved security over the authenticity of the tag.

c. *Direct Control via API Calls*

The third and final method we use to control features on the device is perhaps the most basic and thereby potentially the least effective, namely direct API calls to the feature itself. We utilize this method for the microphone as we find that the permissions needed to mute the microphone through the AudioManager class are not deemed dangerous (see Figure 36). This provides instant state change to mute in a lockdown scenario and until our application releases the microphone state from mute, the mic is largely unavailable to the system. The potential flaw to this design is a situation where another application specifically calls on the OS to release all instances of the microphone in a muted state. Depending on how the OS adjudicates the precedence of that conflict between our application and another the OS determines which application would maintain control over the microphone and its muted state. This noted flaw is not observed in over one dozen attempts to use the microphone through other applications, whereby our application always remains in control of microphone and its muted setting.

Figure 36. Audio manager class



```
//microphone direct lock
AudioManager mAudioManager = (AudioManager)getSystemService(Context.AUDIO_SERVICE);
mAudioManager.setMicrophoneMute(lockdown);
```

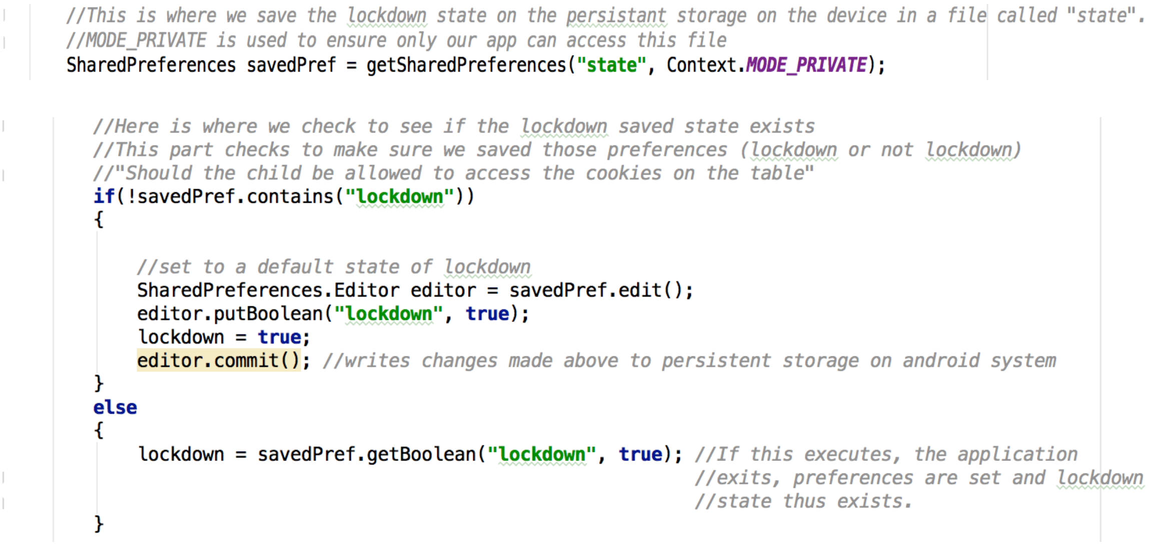
Security application lock for the microphone.

d. *State Related Features*

One of the persistent issues we identify in the early stages of development is the management of the state of the device. We want as few states as possible to prevent situations where complexity overruns the system logic required to manage the state of the device. We have come to the conclusion that we need to use binary logic toggling, and create the variable lockdown=!lockdown to implement this logic. This toggle is located inside the processFinish method that is triggered by the NFC interface. Once the NFC tag is read, the toggle occurs, resulting in the various lockdowns associated with each of the five features.

Recognizing the need for the device to maintain a stored value that it could reference to check itself and reference as a context, we add the code in Figure 37 to the onCreate method to ensure it is built into our application from initiation on the device. Furthermore, we store the value of lockdown in persistent storage and through MODE_PRIVATE make it accessible only to our application, thereby using the system permissions that had previously worked against us to our advantage, by prohibiting other applications from accessing the applications data.

Figure 37. Shared preferences class



```
//This is where we save the lockdown state on the persistant storage on the device in a file called "state".
//MODE_PRIVATE is used to ensure only our app can access this file
SharedPreferences savedPref = getSharedPreferences("state", Context.MODE_PRIVATE);

    //Here is where we check to see if the lockdown saved state exists
    //This part checks to make sure we saved those preferences (lockdown or not lockdown)
    //"Should the child be allowed to access the cookies on the table"
    if(!savedPref.contains("lockdown"))
    {

        //set to a default state of lockdown
        SharedPreferences.Editor editor = savedPref.edit();
        editor.putBoolean("lockdown", true);
        lockdown = true;
        editor.commit(); //writes changes made above to persistent storage on android system
    }
    else
    {
        lockdown = savedPref.getBoolean("lockdown", true); //If this executes, the application
                                                           //exits, preferences are set and lockdown
                                                           //state thus exists.
    }
```

Security Application storage of the device lockdown state. Adapted from: Darkie, and Syed Junaid, "Android Shared Preferences," Sharedpreferences. October 20, 2015, http://stackoverflow.com/questions/23024831/android-shared-preferences-example.

Note that editor.comit writes to storage immediately vice editor.apply, which executes the write command in the background, meaning as the OS decides write time. We want the state written immediately and not as determined by the OS to prevent any race conditions related to state of the device.

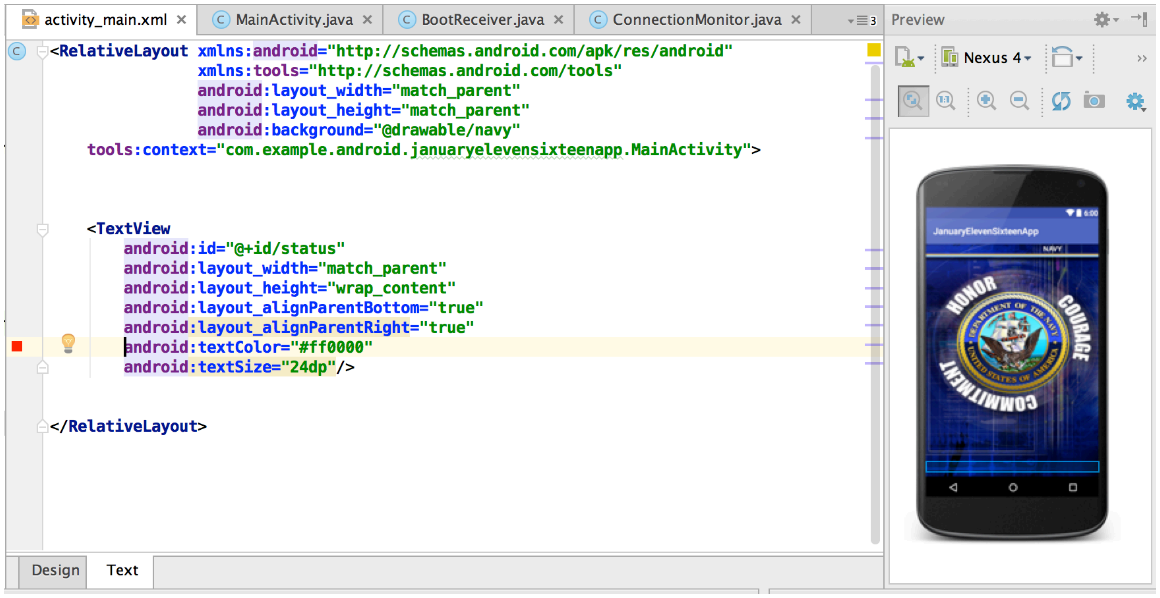
D. DESIGN LAYOUT

As stated throughout Chapter III, and earlier in this chapter, by embracing the Android Studio IDE we are fully immersed in Google's Android Design. The standards and specifications therein provide developers with optimal functionality and esthetically pleasing experiences for their intended users. With that said, for our application we stay as close as possible to Google's design style and layout guidance. For the background, we utilize the Navy ethos "Honor Courage Commitment" image available at www.Navy.mil (see Figure 38). We put that image within a relative layout in order to keep the "layout hierarchy flat, which improves performance."[270] We then add a text view to display the

[270] Google, "Relative Layout," accessed January 21, 2016, http://developer.android.com/guide/topics/ui/layout/relative.html.

system status and aligned it with the bottom of the screen and made the text color red. This provides focused, visually eye catching, immediate feedback to anyone wanting to ascertain the status of the device features, say for example a security manager in a secure space. The status text view is updated and sent to the screen programmatically inside the state change functions in the MainActivity.java file, again, according to Google's principles for optimum performance.

Figure 38. activity_main.xml code for application layout

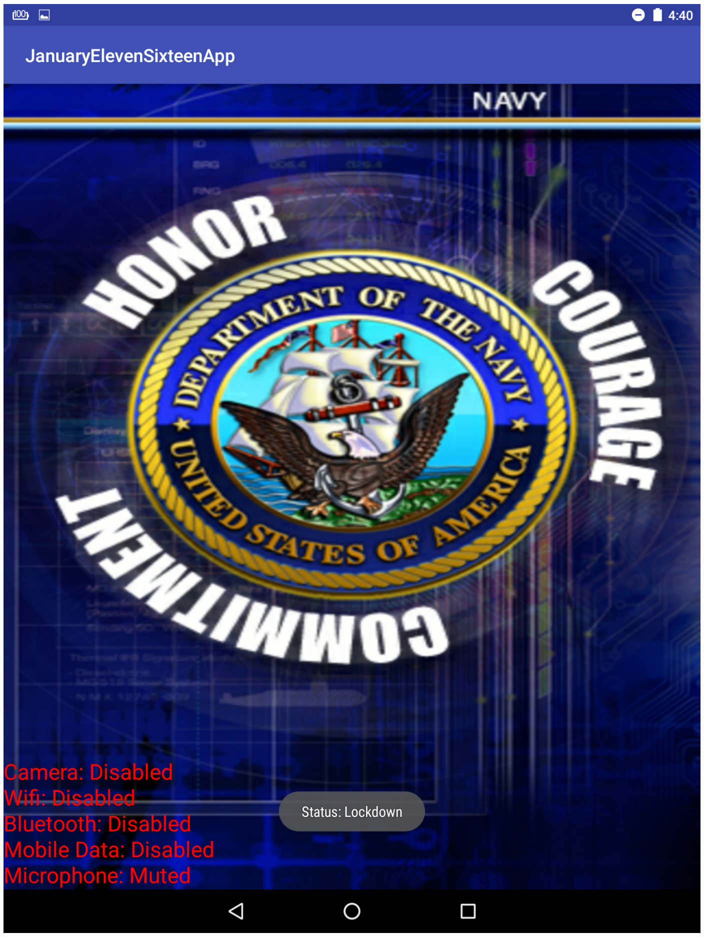


```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
                xmlns:tools="http://schemas.android.com/tools"
                android:layout_width="match_parent"
                android:layout_height="match_parent"
                android:background="@drawable/navy"
    tools:context="com.example.android.januaryelevensixteenapp.MainActivity">


    <TextView
        android:id="@+id/status"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_alignParentBottom="true"
        android:layout_alignParentRight="true"
        android:textColor="#ff0000"
        android:textSize="24dp"/>


</RelativeLayout>
```

E. DEMONSTRATION

The application we developed is pictured as seen on the screens of a Nexus 9 in Figures 39 and 40. One notable takeaway from these images is the simplified structure of the display, whereby a previously approved organizational image, namely the Navy ethos, is used as a background, rather than recreating the wheel so to speak and developing the layout form the ground up. Another notable feature is the toast message in Figure 39 that displays the NFC initiated state change. This provides the user with instant feedback that the NFC has in fact been swiped and as such the state has in fact changed. Also notice the system status in red, and immediately visible to anyone who wishes to monitor the status of the system. The statuses are updated with the java code for the application and indicate the feedback from the receivers or system status.

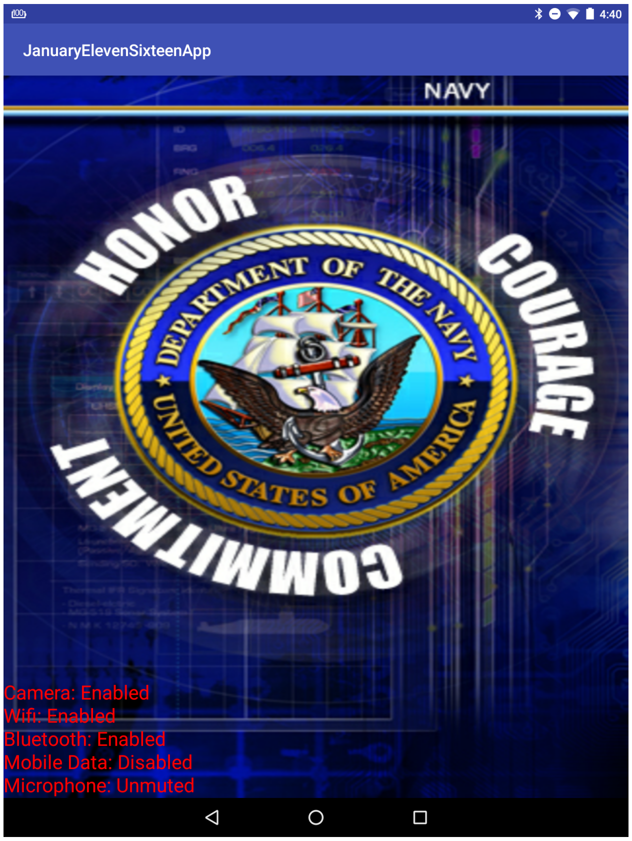
Figure 39. Application on screen in lockdown mode



Camera: Disabled
Wifi: Disabled
Bluetooth: Disabled
Mobile Data: Disabled
Microphone: Muted

Status: Lockdown

Figure 40. Application on screen in unlocked mode



JanuaryElevenSixteenApp

Camera: Enabled
Wifi: Enabled
Bluetooth: Enabled
Mobile Data: Disabled
Microphone: Unmuted

F. TESTING

Based on our software engineering track and associated exposure, we decided early on that we would utilize the Agile testing approach. Between the two of us, we developed over a dozen different applications, with several sub variants of the last application. We progressed from testing code on the built in emulator on Android Studio to actual devices following completion of the NFC portion of the application. Testing took place utilizing SPAWAR provided mobile devices, and principally on a Nexus 9 and Samsung Galaxy S6.

1. Testing Approach

We begin each test cycle by uninstalling previous versions of the application, then initiating the application from Android Studio. This means the latest build variant is loaded on the device for testing. The approach we utilize is to test each feature for lockdown and unlock functionality via NFC tag activation. We continue manipulating the code related to each feature until we achieve lockdown, then subsequent unlock. Once state change is achieved, we test access to the feature by other applications on the device, starting with system applications such as Settings, and then to others such as camera and then finally to third party application attempts to access locked features. Findings are discussed below.

2. Camera

The Device Policy Manger works so seamlessly that controlling this feature is the single easiest portion of developing this application. We literally generate code to create the administrator API, put it inside our NFC broadcast receiver and it completely handles every attempt by the system to access the camera. By default, the camera access is also denied for video capture when in the lockdown state. We identify this here in testing as further evidence that the DOD and DON should approach Google with respect to adding the other features identified earlier in this chapter for lockout via the Device Policy Manager class.

3. Camera Race Condition

A race condition initially existed between calling the function to disable the camera and the installation of the device administration policies. This results in a device administration popup prompt for authorization of the application as an administrator every time the application tries to alter the state of the camera to lockdown. The solution is to leave camera access enabled on application installation, relying on the NFC tag to initiate lockdown on initial swipe by moving the disable function inside the NFC receiver. We apply this NFC receiver solution across all five features that we control, placing all of their disable functions inside the NFC receiver.

4. Wi-Fi

On application start, we toggle Wi-Fi interface to on/off off/on using Wi-FiManager. The act of toggling the Wi-Fi interface results in the OS sending an intent via broadcast, which got picked up by our broadcast receiver. The broadcast receiver is set to detect if supplicant state was to set to INTERFACE_DISABLED, if not, then we set Wi-

FiEnabled to false, triggering WiFi to close anytime an attempt is made to set Wi-Fi to enabled. This results in a permanent state where Wi-Fi interface is inhibited from being enabled, regardless of the state of the device (locked/unlocked). The fix is to create a context based intent filter that is received by our broadcast receiver. This allows us to associate our applications context with the broadcast receiver. Hence when the broadcast receiver onReceive function is called, we have the application context (locked/unlocked), and therefore access all the states associated with the application, allowing us to know what the current state is, allowing us to toggle Wi-Fi, vice simply force it into locked out at all times.

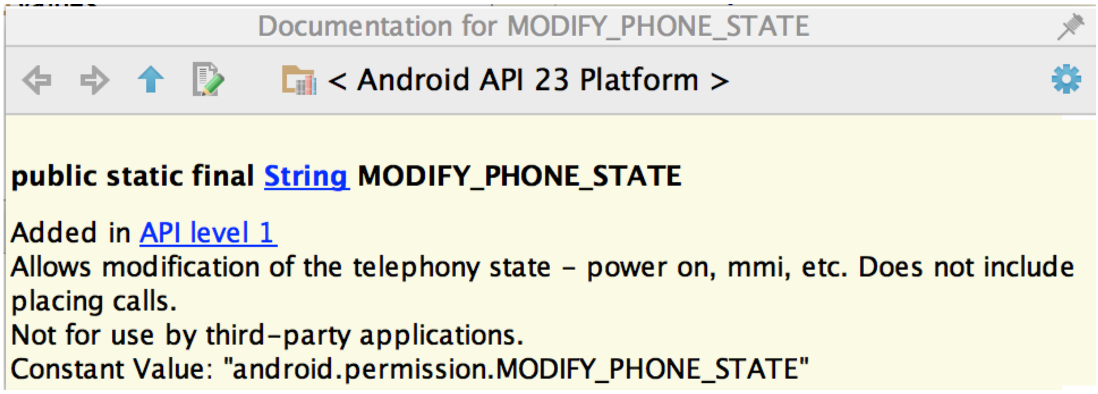
5. Shut Down and Restart

With the app running and in a state of lockdown set we power down the device, then restart. Our observations are very interesting as upon restart, camera access remains disabled due to its device admin policy membership, while Wi-Fi and Bluetooth both regain full functionality and restart. After both services are confirmed up, we re-open our application which immediately disables connection to Wi-Fi and Bluetooth. This means that the state of the application is held within the application itself, so that upon application restart it places the device in the anticipated state (in this case lockdown). We now realize that we need a way to have the application startup with system start-up to prevent a user from utilizing a power down to circumvent the policies enforced by the application. The fix for this problem is to make the application start on boot, using proper boot permissions. The solution as recommended by Google Developer is to implement a new class called BootReceiver, and is henceforth seamlessly integrated into our application. Now even on reboot, the application starts and restores the state of the device on shutdown.

6. Mobile Data and Permissions

Perhaps the most significant finding in our research comes from testing our application across multiple platforms. As stated throughout our research we intend to have our application be as broadly relevant to as many mobile devices used by the widest cross section of sailors as possible. With that said, one very important feature comes about in the most recent Android API update. The loss of programmatic access to the MODIFY_PHONE_STATE permission and method, both of which are required for accessing functionality associated with mobile data. We did not initially observe the change, but rather came across a loss of functionality with respect to locking out device access to mobile data on a relatively new device. We receive the following Android monitor IDE log report when attempting to use andoird.permission.MODIFY_PHONE_STATE: “java.lang.SecurityException: Neither user 10216 nor current process has android.permission.MODIFY_PHONE_STATE.” Through further analysis we find that the permission to access has been removed effective Android API 23 as per Figure 41. The documentation points at MODIFY_PHONE_STATE as “Not for use by third-party applications.” This would include our application and any other application not developed directly by Google for Android, or not a registered system application provided by a mobile device producer or provider (such as Sony, or AT&T).

Figure 41. Android studio API guidance documentation


Documentation for MODIFY_PHONE_STATE
< Android API 23 Platform >

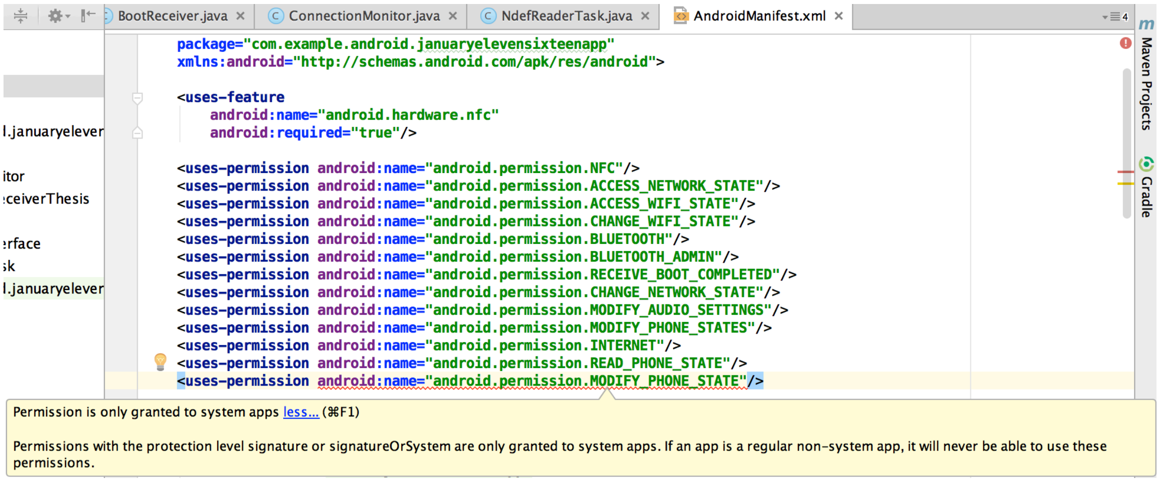
public static final String MODIFY_PHONE_STATE

Added in API level 1
Allows modification of the telephony state – power on, mmi, etc. Does not include placing calls.
Not for use by third-party applications.
Constant Value: "android.permission.MODIFY_PHONE_STATE"

Screenshot from Android Studio IDE

We have made multiple further attempts to access the permissions required to lockout access to mobile data though the AndroidManifest.xml file, but continue to receive the error message noted in the yellow box in Figure 42.

Figure 42. Android IDE automatic editing guidance



```
package="com.example.android.januaryelevensixteenapp"
xmlns:android="http://schemas.android.com/apk/res/android">

<uses-feature
    android:name="android.hardware.nfc"
    android:required="true"/>

<uses-permission android:name="android.permission.NFC"/>
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE"/>
<uses-permission android:name="android.permission.ACCESS_WIFI_STATE"/>
<uses-permission android:name="android.permission.CHANGE_WIFI_STATE"/>
<uses-permission android:name="android.permission.BLUETOOTH"/>
<uses-permission android:name="android.permission.BLUETOOTH_ADMIN"/>
<uses-permission android:name="android.permission.RECEIVE_BOOT_COMPLETED"/>
<uses-permission android:name="android.permission.CHANGE_NETWORK_STATE"/>
<uses-permission android:name="android.permission.MODIFY_AUDIO_SETTINGS"/>
<uses-permission android:name="android.permission.MODIFY_PHONE_STATES"/>
<uses-permission android:name="android.permission.INTERNET"/>
<uses-permission android:name="android.permission.READ_PHONE_STATE"/>
<uses-permission android:name="android.permission.MODIFY_PHONE_STATE"/>
```

Permission is only granted to system apps less... (⌘F1)

Permissions with the protection level signature or signatureOrSystem are only granted to system apps. If an app is a regular non-system app, it will never be able to use these permissions.

IDE provided guidance on the use of the android.permission.MODIFY_PHONE_STATE.

As a result of the updated Android API 23 access to our applications ability to manipulate mobile data access has been removed. This makes the case more strongly for DOD and DON interaction with Google for access to this feature with in the Device Policy Administration class and Administrator API. We provide two potential solutions: (1) have this application deemed a system application by Google or (2) approach Google for further expansion of their Device Policy Manager class and administrator API for inclusion of all 8 features identified in the first part of this chapter. The former likely needs to involve liaison with Google in the form of a Navy representative, which is currently in the works via the secretary of the Navy's Tour with Industry in which Google is an approved partner.[271] The latter option provides more immediate programmatic control of the features we sought to lockout, but would require programmatic development by google to incorporate those features into the associated class. A best case scenario likely involves some hybrid of the two solutions coordinated through NPS under future research.

[271] Department of the Navy, "Secretary of the Navy Tours with Industry," accessed February 15, 2016, http://www.public.navy.mil/bupers-npc/career/talentmanagement/Pages/SNTWI.aspx.

V. SUMMARY AND FUTURE WORK

The rest of this section attempts to answer where these devices can be better enabled to operate in a DOD environment, meet current mobile device requirements, and provide a structure to plan implementation. It is not all-inclusive, but can guide towards reasonable implementation for enabling maintenance records access, unclassified forms, eLearning, or even safety information for jobs and chemicals. We have stepped through what we believe is required from documentation described in Chapter II and is available from Android's device solutions.

A. THE LARGER PICTURE OF BYOD

The application demonstrated in this thesis attempts to tackle the security vulnerabilities presented by the hardware capabilities on the smart device with a lightweight, software solution. We clearly attempt to address the issue of the insider threat by disabling those capabilities of a device that present an opportunity to violate security. However, this is just one small part of a much larger DOD initiative that seems to have a lot of backing but no apparent way forward. That is not to say that DISA and NIST are ignoring the potential in these devices, but so far BYOD has been a big project for the future.

Enterprise solutions enabling reasonably current smart devices are in place, but these devices are still behind the technology curve in most cases. Examining DISA's approved product list supports this claim.[272] While it is possible to now use an Android enabled Samsung Galaxy S6, it is still only allowed on a prior operating system version. The iOS devices are still limited to iPhone 5 variants and are only allowed up through iOS 8. Since iOS 9 is now the standard, procuring devices with the older operating system creates a challenge to these devices. When purchasing new devices from Apple, it

[272] Defense Systems Information Agency keeps a list of currently approved products on their website. This is supported with dropdowns to search by device type. When visiting the site (https://aplits.disa.mil/processAPList.action) a user should select multifunction mobile devices for all vendors to get a full list of currently approved devices and view the valid operating system and notifications associated with each.

is the company's policy that the most current version of the operating system be on the device.

B. RECOMMENDED FEATURES

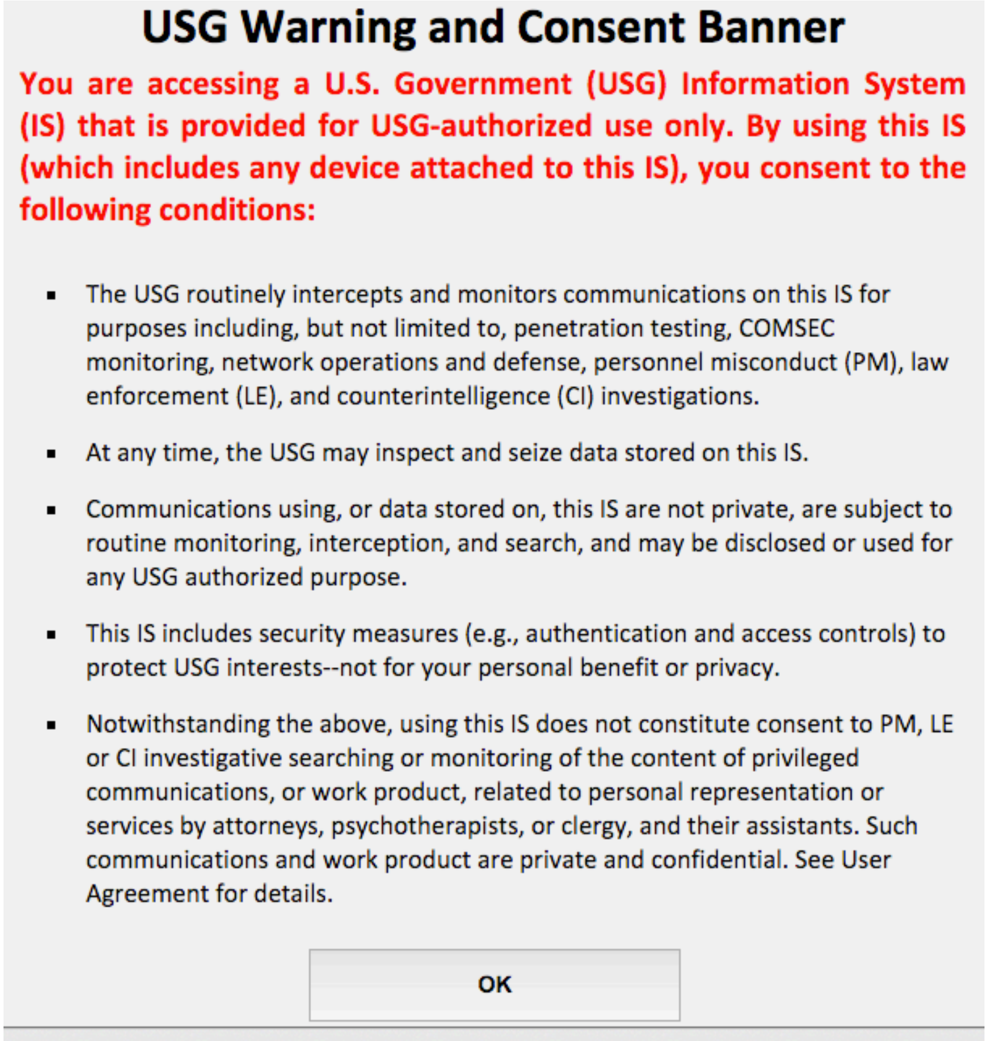
Our application touches on one small piece of securing a device to be carried about a Navy ship. The potential for future work and making this an implementable solution cannot be overstated. The following areas are either required by previous policies referenced are are recommended to grow the application to play a bigger role in BYOD.

1. Authorization Banners

Access to a DOD system or unclassified devices used in a DOD facility is generally initiated by accepting a warning banner. This banner informs the user of proper use of the system, what is not acceptable, and stands as notification that accessing the network guarantees no privacy if data is transferred over the network. Additionally, it has a user interaction that is required to proceed. This is generally a simple click of an "OK" button or clicking a tick box and then proceeding. In any scenario where this banner exists, it does require active user acknowledgement to proceed.

A pop up of this nature could be built into the application and should exist prior to being implemented in any fashion. The pop up should happen after scanning the NFC tag and should immediately required the user to hit ok before proceeding. A time limit to accept does not need to be established for the banner, but the application should keep track of whether or not this banner was accepted (this would be a simple Boolean state change) before actually releasing any other activity on the phone. To be noted, the policy put in place by the NFC tag scan should be immediately implemented, regardless of whether the banner has been acknowledged. The phone should still have all required hardware and transfer methods disabled, even if the user forgot to click ok on a policy banner. An example of a DOD email banner text is given in Figure 43.

Figure 43. An example consent to use banner from a military email exchange server



USG Warning and Consent Banner

You are accessing a U.S. Government (USG) Information System (IS) that is provided for USG-authorized use only. By using this IS (which includes any device attached to this IS), you consent to the following conditions:

- The USG routinely intercepts and monitors communications on this IS for purposes including, but not limited to, penetration testing, COMSEC monitoring, network operations and defense, personnel misconduct (PM), law enforcement (LE), and counterintelligence (CI) investigations.
- At any time, the USG may inspect and seize data stored on this IS.
- Communications using, or data stored on, this IS are not private, are subject to routine monitoring, interception, and search, and may be disclosed or used for any USG authorized purpose.
- This IS includes security measures (e.g., authentication and access controls) to protect USG interests--not for your personal benefit or privacy.
- Notwithstanding the above, using this IS does not constitute consent to PM, LE or CI investigative searching or monitoring of the content of privileged communications, or work product, related to personal representation or services by attorneys, psychotherapists, or clergy, and their assistants. Such communications and work product are private and confidential. See User Agreement for details.

An example of a government system access warning dialogue box. This requires acknowledgement by the user before any access is granted. A similar box should be adapted before our application is used. The text should mention specifically what the application blocks and that subverting any of the applications normal operation is a violation of secure BYOD policy. Adapted from unclassified military mail access site: https://web-mont05.mail.mil/my.policy.

2. Separate Memory Space

The Android operating system allows applications to save files within an application such that the files cannot be accessed by the user outside of the application and no other application can reach in and pull the information out.[273] Management of

[273] Storage of files within an application and device internal storage management is described in the *Android Developer Guide* at http://developer.android.com/guide/topics/data/data-storage.html.

storage on a device can give some powerful flexibility when it comes to an implementable BYOD policy. Enterprise solutions already create a partition in memory for access to classified and unclassified information and official access to email and contacts. Files are accessed and maintained within the application's allocated storage, reducing the possibility of spillage. Setting up the internal storage is not a difficult task for an application programmer, but it would absolutely be a requirement for any BYOD application that designed to access official files on a local network or pull emails from a DOD exchange server.

With reference to future applications, we would like to see storage space set aside for saving emails, ship's policy instructions and publications, and access to learning requirements. This information, saved locally, would allow greater flexibility for those users that routinely need information, possibly outside of access to a network. An example of where this would be useful is in Force Protection. Generally, before pulling in to a foreign country's pier, the ship's leadership will receive force protection updates and the pier laydown (how the pier is physically configured along with security structures). As a Force Protection Officer (FPO), it would be a benefit to be able to walk the pier once a ship has pulled in, compare current conditions with ship's instructions and Navy force protection policy, and adjust maps so that the host/husbanding agent can ensure the greatest level of protection. Accessing this information before leaving the ship's network and being able to carry it on a manageable device instead of multiple binders would give the FPO greater flexibility and keep references at his fingertips. Creating an application with allocated storage space would allow for this feature.

Since Android already separates files saved within an application, access to the data from outside of that application's sandbox is controlled and is not possible. There would have to be limits to what is kept on the phone, but that falls under policy and could be set up within the application. Additionally, establishing file size limits will help manage storage space and keep users aware of the items that are saved on their phone. Removing older, no longer needed data should be a requirement so that only those files and records needed to do a current job are kept inside the application.

3. Network White Listing

Building a ship’s network to access additional features will be discussed later in this chapter, but the utility of accessing a list of approved networks should not be overlooked. Security is the goal of our application and we have demonstrated those features that can be controlled in an Android environment. It is worth noting that programmers are also provided methods by which they can limit access to networks or provide a list of approved networks. Putting this information directly into the application would require oversight and should not be changed at the user level without a security manager’s approval, but it can be built in. Further discussion of why this is useful will be discussed in the section about a ship’s wireless network.

4. Mobile Data Solution

Since Android has removed the ability to control mobile data, and it is unknown if a creative solution to shut this feature down exists, then some exploration on this topic is warranted. Android removed the access to this feature under the auspices of preventing applications from turning on or increasing mobile data usage. A lot of control is however currently offered to users to monitor data usage, which applications are using the most data, and mobile data can be restricted app by app instead of only turning it off or on. This is very handy for users. If access to the way this works can be gained via code, then the application could possibly just set all data moving through apps to zero. If a solution is not provided by Google, then this is a communication consideration that may not be able to be secured at the application level.

5. Report Space Casualty

This particular recommendation is a long-distance hope that makes sense based on our combined time on surface ships and submarines. Once an application and intranet has progressed to where policy appropriately manages mobile devices on the unit, logging exists for which devices are permitted on the network, and users have been trained on appropriate use, it is not unreasonable to place a button on the app that can report smoke, flooding, a personnel accident, etc. This should never be used when traditional reporting mechanisms are available, but in a connected ship it could be beneficial. Consider a situation in which a sailor has been trapped or injured and is unable to reasonably move to where he can report his condition or situation. If this sailor had a mobile device on him with access to a connection, he could hit a button that reported to a server that would notify someone sitting watch in engineering and on the quarterdeck, as appropriate. The logging of such notifications and on a server would minimize the chance for abuse to levels similar to internal communication reporting mechanisms. This simple addition could provide a mechanism for injured or trapped sailors to report their condition prior to being discovered by someone making rounds.

6. Plug Ins

When we discussed this idea for an application that locks down a device with SPAWAR, a Marine listening to the conversation was quickly interested. He related a story where on deployment in Afghanistan he and his men were having issues with the maps they were taking when leaving base. Because of how dynamic the environment was and how often they were diverted, carrying the necessary paper maps for such a large area was almost impossible. He worked through his chain of command, eventually getting an admiral's permission, to use iPads to load area maps. This is not normally acceptable and the devices are not permitted. However, his environment necessitated an innovative approach to the problem.

When we mentioned how we could shut down components of the device to eliminate some security concerns he asked about the feasibility of adding access to other information, based on his experience with the maps. This led us to consider adding a plug in feature for the application. We imagine an application that, once a device is secure, provides a drop down or menu to available features. If a sailor wishes to load ship instructions, the PDF can open up in the application with a PDF viewer component. If a sailor wishes to complete training, then a lite, in-app browser opens the page on a local server to finish and report the training. If a sailor is conducting maintenance, then the appropriate MRC can open. Adding plug ins for additional features would not be as straight forward, and would require additional oversight to ensure they were not bypassing security features, but could be a manageable, flexible addition.

C. SEPARATE TECHNOLOGY

In addition to the recommended changes above, solutions off the device are also proposed. When added to a function application that locks down features and has the capability to be used by the sailor, the following pieces of technology will help in providing a more secure and useful environment.

1. Ship's Wireless Network

As the application stands, we attempt to disable as many ways to transfer data as possible. This demonstrates that data transfer methods can be controlled. It is not absolutely necessary to disable the Wi-Fi on the device, and in the future it would be beneficial to have Wi-Fi turned on. This would enable access to a local network that contained ship's databases, document repositories, and online training. Having a Wi-Fi network white list could enable the application to choose to connect to the ship's intranet while ignoring other incoming connection requests. This would also allow moving the Internet control from being a smart device consideration to a network setup consideration, reducing greater app requirements.

When a ship's intranet is set up, the network administrator can easily set up folders that could be accessed from an application. We are considering the ability to pull Material Safety Data Sheets (MSDS) for hazardous materials, Maintenance Requirement Cards (MRC) for gear, and even access to ship's publications, email, or the lightweight version of Navy Learning. The ship's security manager could operate a database that is checked any time a phone gets on the network, verifying a device is registered and its user has been trained on appropriate usage. A network no only increases the amount that can be done with the device, but gives a greater option for device management to ship's leadership.

Being able to complete training requirements from a tablet or phone while sitting on the mess deck would reduce the time sailors spend waiting for a computer to open up and would increase how quickly commands can report that a training requirement is completed. Additionally, a network administrator could configure the network to refuse and block all inbound/outbound connections not specifically associated with NKO. In this way, if a high-speed connection were available, access to training could be increased beyond what is available on the ship's server version of NKO.

The flexibility to use a device appropriately will increase when a network communication option is offered. However, policy must be established to outline smart use and network setup. Restrictions against internal ship networks will have to be adjusted to accommodate routine use of a wireless connection. Placement of access points must be considered and power must be adjusted so that they cannot be accessed outside of desired spaces, or more importantly, inside off-limits spaces. Additionally, the previously mentioned banner must be incorporated into the app and it must mention that a DOD or Navy network is being accessed. Implementing a ship's Wi-Fi network for smart devices would not be difficult in practice, but would require substantial planning and documentation so that it follows Navy policy and best practices.

2. Publication Server

Sitting on the wireless network can be access to various pieces of information. One of the frustrations of working on a ship is the need to get regular access to ship's documents and publications. Unclassified, FOUO documents are referenced regularly while in various spaces throughout the ship. Having the ability to access a repository of manuals, ship's instructions, the Plan of the Day, and other documents as leadership sees fit would enable a sailor to more easily seek documented guidance. Given the limited number of computers on a ship this could provide a large amount of flexibility and would be references at the fingertips of leadership and junior sailors alike.

3. Ship's Database and Management

Chapter IV discussed in depth an HTML implementation of a security manager interface. This lightweight example allows for logging of devices that are permitted on the ship, can provide when settings are changed on the phone, and could push updates to the main screen of the sailors' devices once configured. These updates could be notifications of required training or that their device is no longer permitted until a security update has been performed to the application. The security manager would also use this interface to configure and write new NFC tags for use in accordance with the CO's standing orders and general guidance.

4. Space Alarms for Mobile Devices

One of the biggest reasons we want to write a software solution to disable hardware capabilities is because of how often we hear stories of a sailor accidentally going into Combat, Radio, or some other restricted space with a mobile device. These will be in the pocket of the sailor, and their presence in these spaces represent a significant security issue. By the letter of current security requirements, the command should send a self-report each time this occurs, and the device is supposed to be taken and examined to ensure no classified data is on the device. While our application would not remove the self-reporting requirement, it can definitely provide amplifying information on the status of the device at the time the violation occurred.

For example, consider the following: a mobile device in Combat for the duration of a scenario in which classified actions and preplanned responses to enemy actions are occurring. Sending a message that details the scenario and that the device was present creates a difficult situation for the CO and the sailor but is a requirement. If the CO had the ability to note in his message that the device had been locked down in accordance with policy, the sailor had received the training prior to entering the space (but clearly requires remediation), and no data had been recorded to the device or left the device should relieve some concern. However, what if none of that were necessary? What if as soon as a sailor entered a space, a sniffer picked up his mobile device, which is searching for a Wi-Fi signal, and alerted leadership that a device had entered the space?

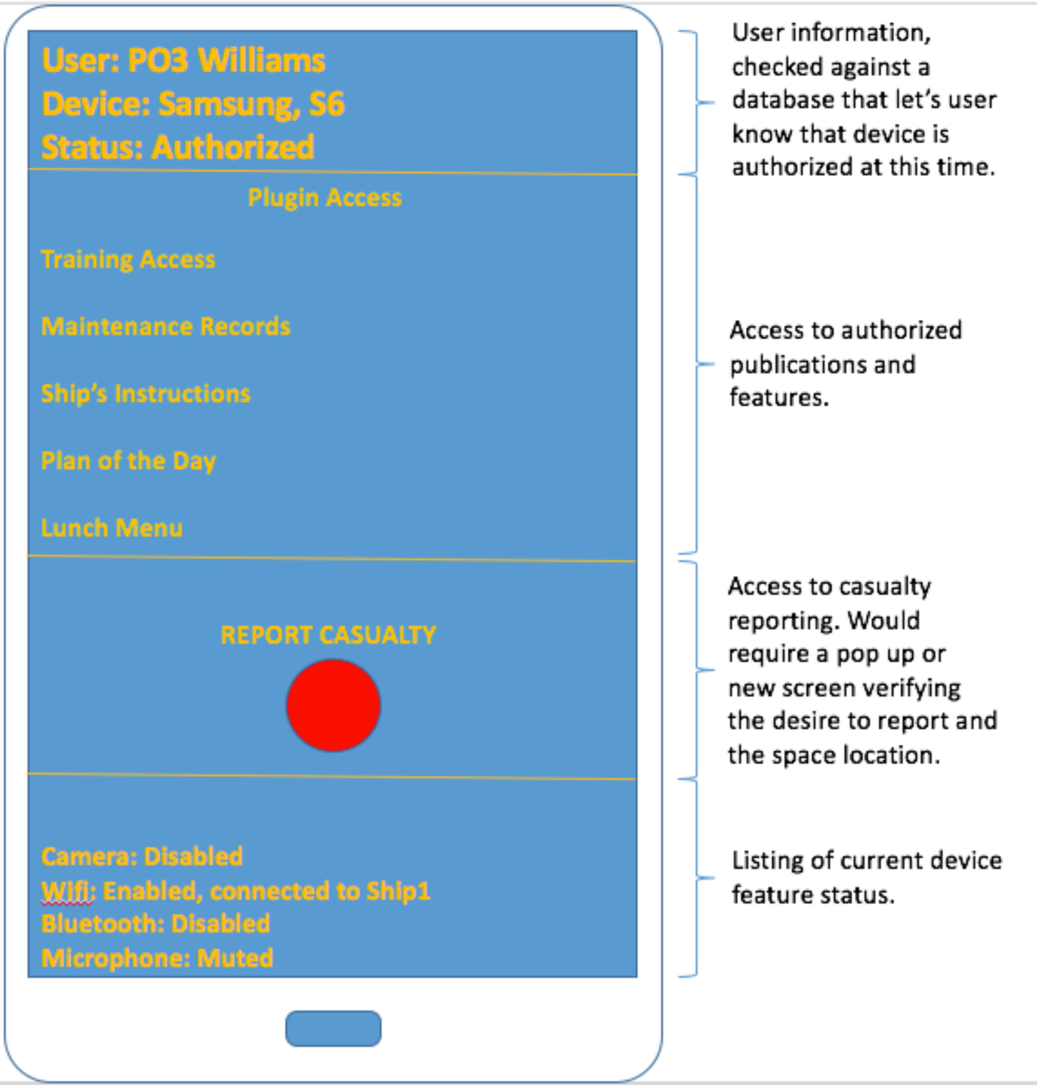
There are multiple commercial solutions that would easily allow such a situation to exist. Devices that detect cell phones are not new, and sensitivity to the various communication methods are device dependent. However, installing and tuning a detection system to an appropriate power setting such that it can sense a mobile device in a space such as Combat without providing false alarms from devices walking in proximity to the space is not a terribly difficult task. It is our recommendation that as policy adjusts for BYOD, as an application is built, and as a ship's network is managed and grows appropriately that these devices be installed in off-limits spaces for quick notification and reaction of a device entering.

D. FUTURE WORK WRAP-UP

The development of this application allows us to look inside a wide range of publications and standard practices for mobile devices. The thought of minimizing the insider threat drove initial research but expanded into the realm of BYOD. Looking at how much this is discussed by high-level leadership within the DOD and DISA while having very little published on implementation, we feel it is important to provide a small piece for future work. For example, if the goal is to develop an eLearning capability afloat such that sailors can use their personal devices, then future work could focus on going that direction. Security mechanisms should be built using the OS developer APIs and policy should be developed to guide training and implementation.

Without picking a single goal for testing how this will work, getting that goal supported by the DOD and DISA, and providing oversight from security policies BYOD will continue to simply be a goal. If sailors can check into a ship, have their devices set up to operate in the ship's environment, and easily transition to and from that environment then the Navy will have provided something significant for the sailors and taken a smart, active approach at reducing risks and practices dangerous to security. An example of a screen with some of the discussed features is in Figure 44. It is our belief that this application is a starting point for discussion on moving BYOD and ship's security from a future discussion to a reality.

Figure 44. An example of a more developed application screen



User information, checked against a database that let's user know that device is authorized at this time.

Access to authorized publications and features.

Access to casualty reporting. Would require a pop up or new screen verifying the desire to report and the space location.

Listing of current device feature status.

This model of an application home screen takes into consideration how the device would look to a sailor if the future work recommendations were implemented. All of the features discussed for future work are implementable and, with policy, create a useful application for sailor's operating on Android smart devices.

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF REFERENCES

Commander Navy Cyber Forces. *Commander's Cyber Security and Information Assurance Handbook* (COMNAVCYBERFORINST 5239.2A). Rev 2. Washington, DC: Department of the Navy, 2013.

Danberry, Michael J. "MilitaryCAC's Information on Using Your CAC with Your Mobile Device including AKO Email." 'January 13, 2016. https://militarycac.com/mobile.htm.

Defense Information Systems Agency. *Commercial Mobile Device (CMD) Policy Security Technical Implementation Guide (STIG)*. March 12, 2013. https://www.stigviewer.com/stig/commercial_mobile_device_cmd_policy/2013-03-12/.

———. *Mobile Policy Security Requirements Guide*. October 10, 2012. www.stigviewer.com/stig/mobile_policy_security_requirements_guide/2012-10-10/.

———. *Mobile Policy Security Requirements Guide*. January 24, 2013. https://www.stigviewer.com/stig/mobile_policy_security_requirements_guide/2013-01-24/.

———. *Mobile Policy Security Requirements Guide*. July 13, 2013. https://www.stigviewer.com/stig/mobile_policy_security_requirements_guide/2013-07-03/.

———. *Mobile Policy Security Requirements Guide*. October 10, 2012. www.stigviewer.com/stig/mobile_policy_security_requirements_guide/2012-10-10.

———. *Smartphone Policy Security Technical Implementation Guide*. February 2, 2012. www.stigviewer.com/stig/smartphone_policy/2012-02-02/finding.

———. *Smartphone Policy Security Technical Implementation Guide*. October 9, 2012. www.stigviewer.com/stig/smartphone_policy/2012-10-09/.

DENSO ADC. *QR Code Essential* (White Paper R1f). 2011. http://www.nacs.org/LinkClick.aspx?fileticket=D1FpVAvvJuo=&tabid=1426&mid=4802.

Department of Commerce, and National Institute of Standards and Technology. *Security and Privacy Controls for Federal Information Systems and Organizations* (SP800-53r4). Rev. 4. Gaithersburg, MD: Department of Commerce, and National Institute of Standards and Technology, 2013.

Department of Defense. *Interoperability of Information Technology (IT), Including National Security Systems (NSS)* (DOD Instruction 8330.01). Washington, DC: Department of Defense, 2014. https://acc.dau.mil/adl/en-US/706841/file/77077/DOD%20-%20Instruction,%20DoDI%208330.01,%20Interoperability%20of%20IT%20and%20NSS,%2021%20May%202014.pdf.

Department of Defense, Chief Information Officer. *Commercial Mobile Device Implementation Plan*. Washington, DC: Department of Defense, 2013. http://archive.defense.gov/news/DoDCMDImplementationPlan.pdf.

———. *Cybersecurity* (DOD Instruction 8500.01). Washington, DC: Department of Defense, 2014. http://www.dtic.mil/whs/directives/corres/pdf/850001_2014.pdf.

———. *Enterprise Mobility 2008*. Washington, DC: Department of the Navy, 2008.

———. *Optimizing Use of Employee Information Technology (IT) Devices and Other IT to Achieve Efficiencies* [memorandum]. Washington, DC: Department of Defense 2012.

———. *Risk Management Framework (RMF) for DOD Information Technology (IT)* (DOD Instruction 8510.01). Washington, DC: Department of Defense, 2014.

Department of Defense, Mobility Program Management Office. *Memorandum for DOD Mobility Supported Devices*. Fort Meade, MD: DOD Mobility Program Management Office, 2015.

Department of the Navy, Chief Information Officer. *Amplification Guidance for Purchase and Installation of Personal Electronic Device Smart Card Readers* (281919Z JAN 09). Washington, DC: Department of the Navy, 2007.

———. *DON Security Guidance for Personal Electronic Devices* (202041Z AUG 07). Washington, DC: Department of the Navy, 2007. http://www.doncio.navy.mil/uploads/0128BAA54339.pdf.

De Rosa, Cathy, Joanne Cantrell, Matthew Carlson, Peggy Gallagher, Janet Hawk, and Charlotte Sturtz. *Perceptions of Libraries, 2010, Context and Community*. Dublin OH: Online Computer Library Center, 2010.

Dorney, Liam J., Elmas D. Kadir, Sellers Kristin, Jerel W. Yam, and Abdullah Yilmaz. "CS3004 Project 1: Mobile Security at Sea" Class paper, CS3004, Naval Postgraduate School, 2015.

Fox News. "Sailor Faces Charges after Photos of Navy Attack Sub Found on Cellphone." August 3, 2015. http://www.foxnews.com/us/2015/08/02/sailor-faces-charges-possessing-photos-navy-attack-sub-on-cellphone.html.

Goodin, Dan. “950 Million Android Phones Can Be Hijacked by Malicious Text Messages.” Arstechnica. July 27, 2015. http://arstechnica.com/security/2015/07/950-million-android-phones-can-be-hijacked-by-malicious-text-messages/.

Heggestuen, John. “One in Every 5 People in the World Own a Smartphone, One in Every 17 Own a Tablet.” *Business Insider*, December 15, 2013. http://www.businessinsider.com/smartphone-and-tablet-penetration-2013-10.

Hurst, Nathan. “IPhone Separation Linked to Physiological Anxiety, Poor Cognitive Performance, MU Study Finds.” *MU News Bureau*, January 8, 2015. http://munews.missouri.edu/news-releases/2015/0108-iphone-separation-linked-to-physiological-anxiety-poor-cognitive-performance-mu-study-finds/.

Larter, David. “Sailor Faces Charges for Submarine Photos on Cellphone.” *USA Today*, August 1, 2015. http://www.usatoday.com/story/news/nation/2015/08/01/sailor-faces-charges-submarine-photos-cellphone/31005689.

Lennon, Elizabeth. “ITL Issues and Guidelines for Managing the Security of Mobile Devices.” *ITL Bulletin*, July 2013. http://csrc.nist.gov/publications/nistbul/itlbul2013_07.pdf.

Madlmayr, Gerald. “NFC Development & Consulting.” NFC Development Consulting. March 8, 2011. http://www.nfc.cc/technology/nfc/.

Miller, Travis, Jerel Yam, and Liam Dorney. “An Examination of Malware Interactions in the Android OS.” Class paper, CS3070, Naval Postgraduate School, March 2015.

Morris, Brian. “Are QR Codes Thriving or Dying?” Business 2 Community. May 21, 2015. http://www.business2community.com/marketing/qr-codes-thriving-dying-01228016#DoWpfBHtY3rYivO3.97.

Ms. Smith. “Black Hat: It’s Not ‘‘Tricky’ for Hackers to Turn Your Phone into a SpyPhone.” Network World. August 1, 2013. http://www.networkworld.com/article/2225081/microsoft-subnet/black-hat--it-s-not--tricky--for-hackers-to-turn-your-phone-into-a-spyphone.html.

Murach, Joel. *Murach’s Android Programming*. Fresno, CA: Mike Murach & Associates, 2013.

National Institute of Standards and Technology. *Security and Privacy Controls for Federal Information Systems and Organizations* (SP 800–53r4). 2013. http://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-53r4.pdf.

Quirolgico, Steve, Jeffrey Voas, Tom Karygiannis, Christoph Michael, and Karen Scarfone. *Vetting the Security of Mobile Applications* (SP-800-163). Washington, DC: National Institute of Standards and Technology, 2015.

Phillips, Bill, and Brian Hardy. *Android Programming: The Big Nerd Ranch Guide*, Vol. 2. Atlanta, GA: Big Nerd Ranch, 2015.

Rice, Kim. "DOD Mobility, Presentation." Defense Systems Information Agency. June 17, 2015. http://www.disa.mil/~/media/Files/DISA/News/Conference/2015/Secure_MobilityRice.ashx.

Seidman, Tony. "Barcode History: Barcodes Sweep the World." Barcoding Incorporated. Accessed January 20, 2016. http://www.barcoding.com/information/barcode_history.shtml.

Smith, Aaron. *U.S. Smartphone Use in 2015*. Washington, DC: Pew Research Center 2015. http://www.pewinternet.org/2015/04/01/us-smartphone-use-in-2015/.

Souppaya, Murugiah, and Karen Kent. *Guidelines for Managing the Security of Mobile Devices in the Enterprise* (SP800-124r1). Rev 1. Washington, DC: National Institute of Standards and Technology, 2013.

Weiss, Josef. "STIG Alerts (by CAT)." Tenable Network Security, December 18, 2014. https://www.tenable.com/sc-dashboards/stig-alerts-by-cat.

Wright, Joshua. "Dispelling Common Bluetooth Misconceptions." Security Laboratory: Wireless Security. September 19, 2007. http://www.sans.edu/research/security-laboratory/article/bluetooth.

Zhang, Veo. "Hacking Team RCSAndroid Spying Tool Listens to Calls; Roots Devices to Get In." *TrendLabs Security Intelligence Blog*, July 21, 2015. http://blog.trendmicro.com/trendlabs-security-intelligence/hacking-team-rcsandroid-spying-tool-listens-to-calls-roots-devices-to-get-in/.

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California